US 20020143864A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2002/0143864 A1

Page et al. (43) Pub. Date: Oct. 3, 2002

(54) INFORMATION RETRIEVAL SYSTEM

(75) Inventors: David Richard Page, Romsey (GB); Birgit Schmidt-Wesche, Winchester (GB); Jonathan Stone, Romsey (GB)

Correspondence Address:
IBM Corp
IP Law Dept T81/503
3039 Cornwallis Road
P.O. Box 12195
Research Triangle Park, NC 27709-2195 (US)

(73) Assignee: International Business Machines Corporation, Armonk, NY (US)

(21) Appl. No.: 09/877,152

(22) Filed: Jun. 8, 2001

(30) Foreign Application Priority Data

Mar. 30, 2001 (GB) 0107953.2

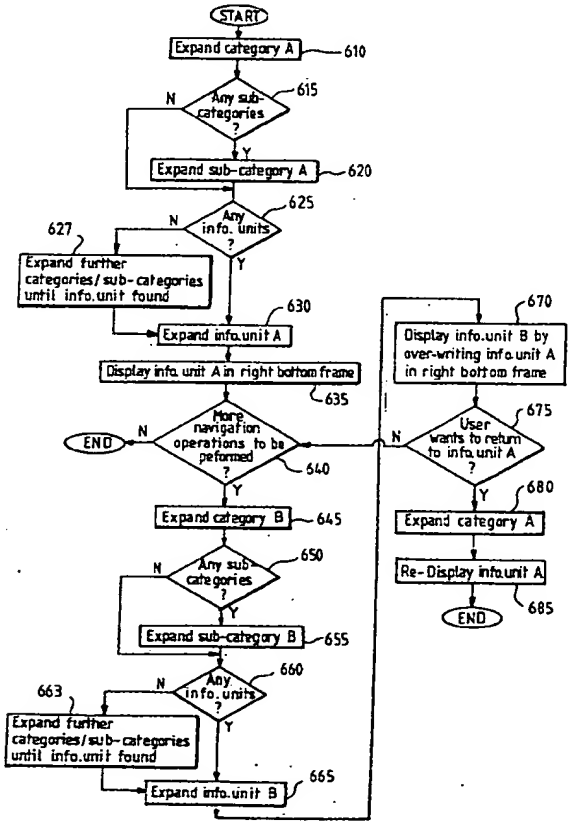
Publication Classification

(51) Int. Cl.[7] G06F 15/16; G06F 17/30; G06F 7/00

(52) U.S. Cl. 709/203; 707/3

(57) ABSTRACT

A method of retrieving information by navigating within a web browsing session. The information is stored on a server in a hierarchical tree comprising a root node, top-level child nodes representing information categories and leaf nodes representing information sub-categories. Each of the nodes has an associated information unit. Initially, an information unit associated with the root node is displayed in an information space. A user then performs navigation operations from the root node, by selecting a first top-level child node, traversing the leaf nodes and selecting a first leaf node. A first information unit associated with the first leaf node is subsequently displayed in the information space. This first information unit is stored, the navigation operations are repeated on a second top-level child node until a second information unit is displayed in the information space. The user then re-selects the first top-level child node and in response to the re-selection the first information unit is automatically re-displayed in the information space.



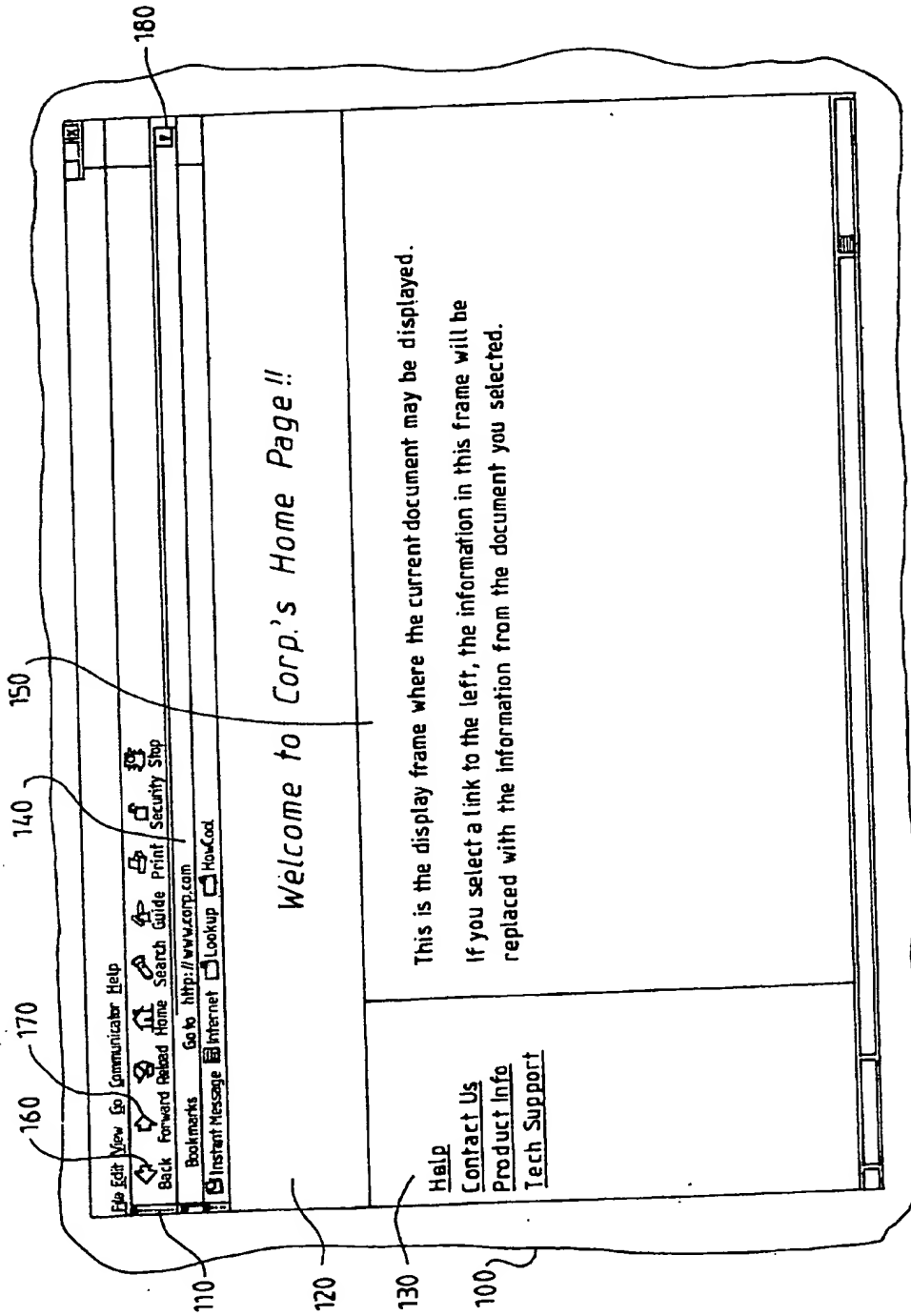


FIG. 1 PRIOR ART

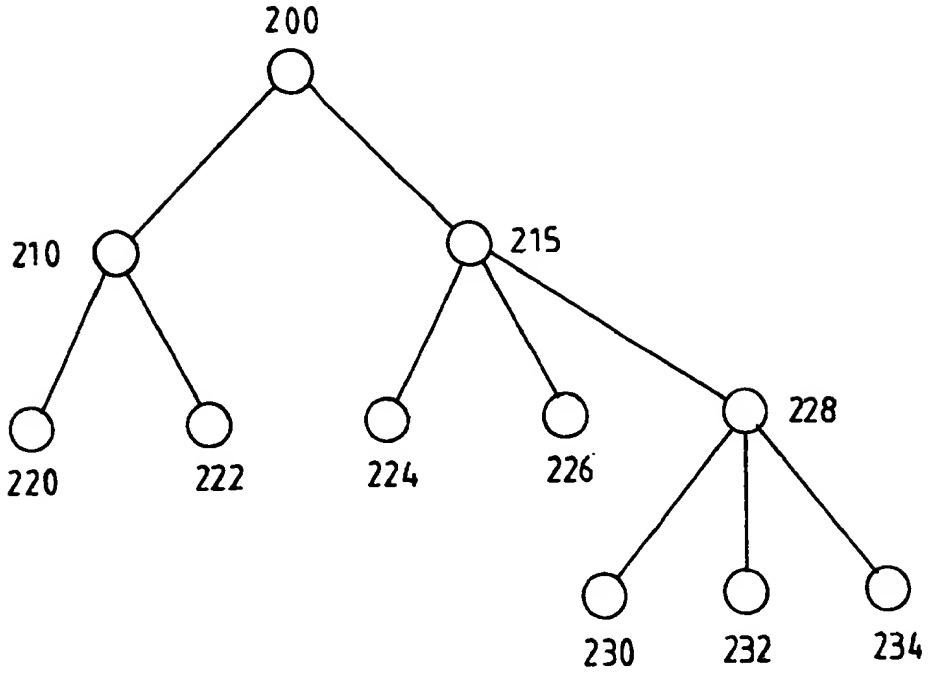
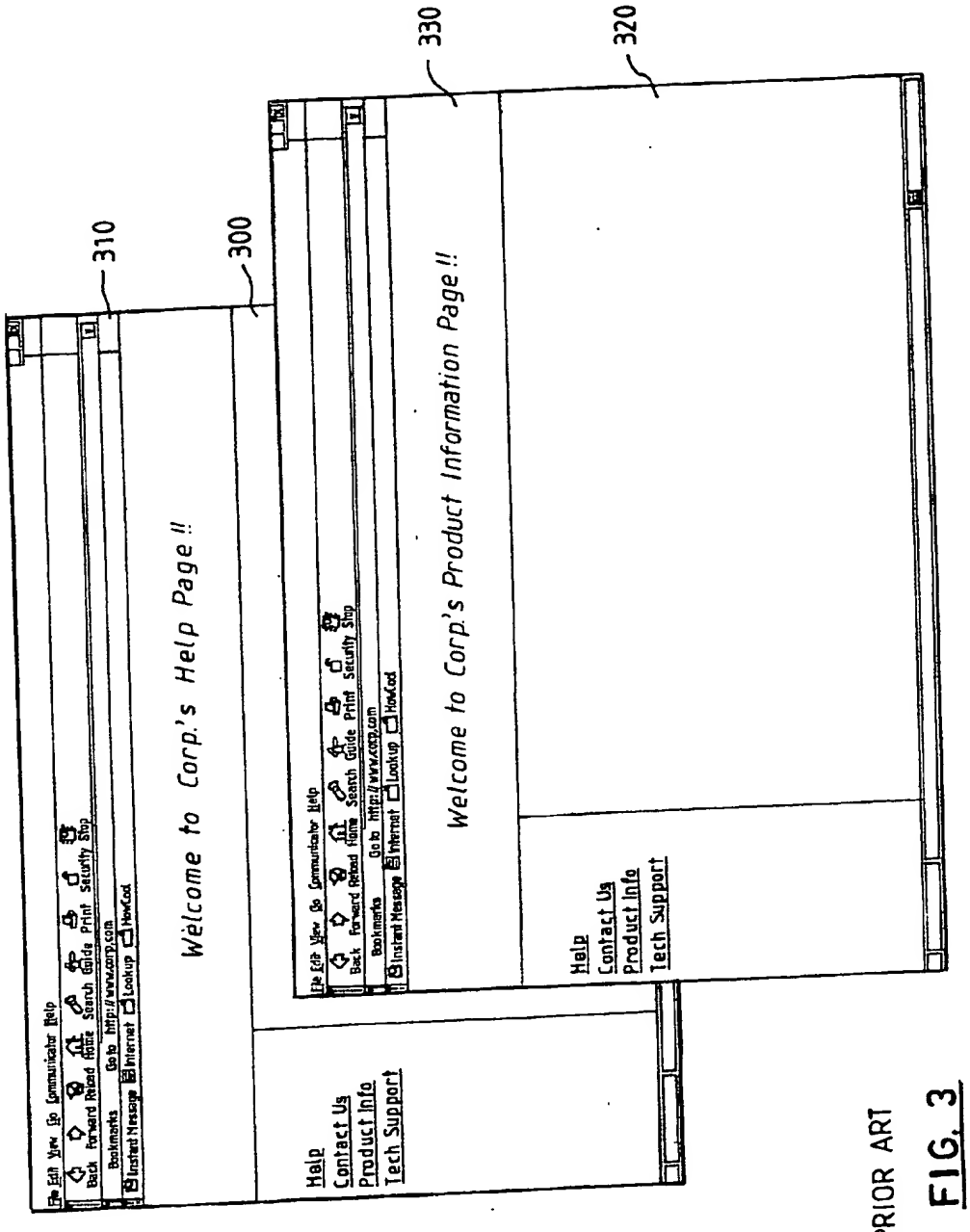


FIG. 2 PRIOR ART



PRIOR ART

FIG. 3

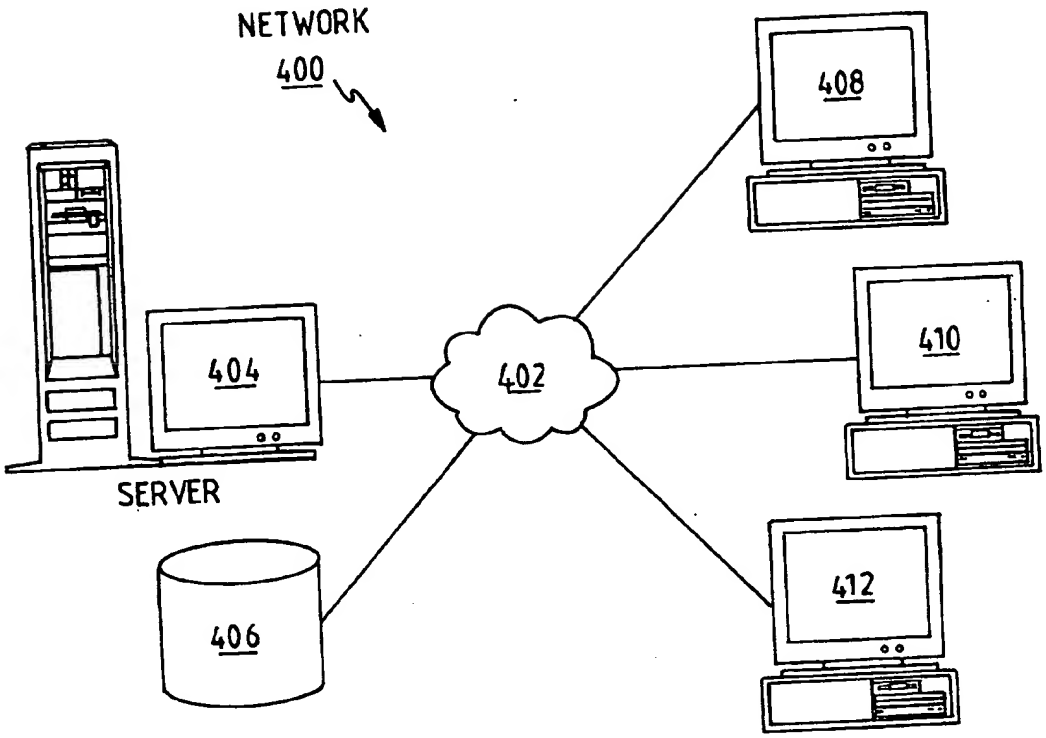


FIG. 4 PRIOR ART

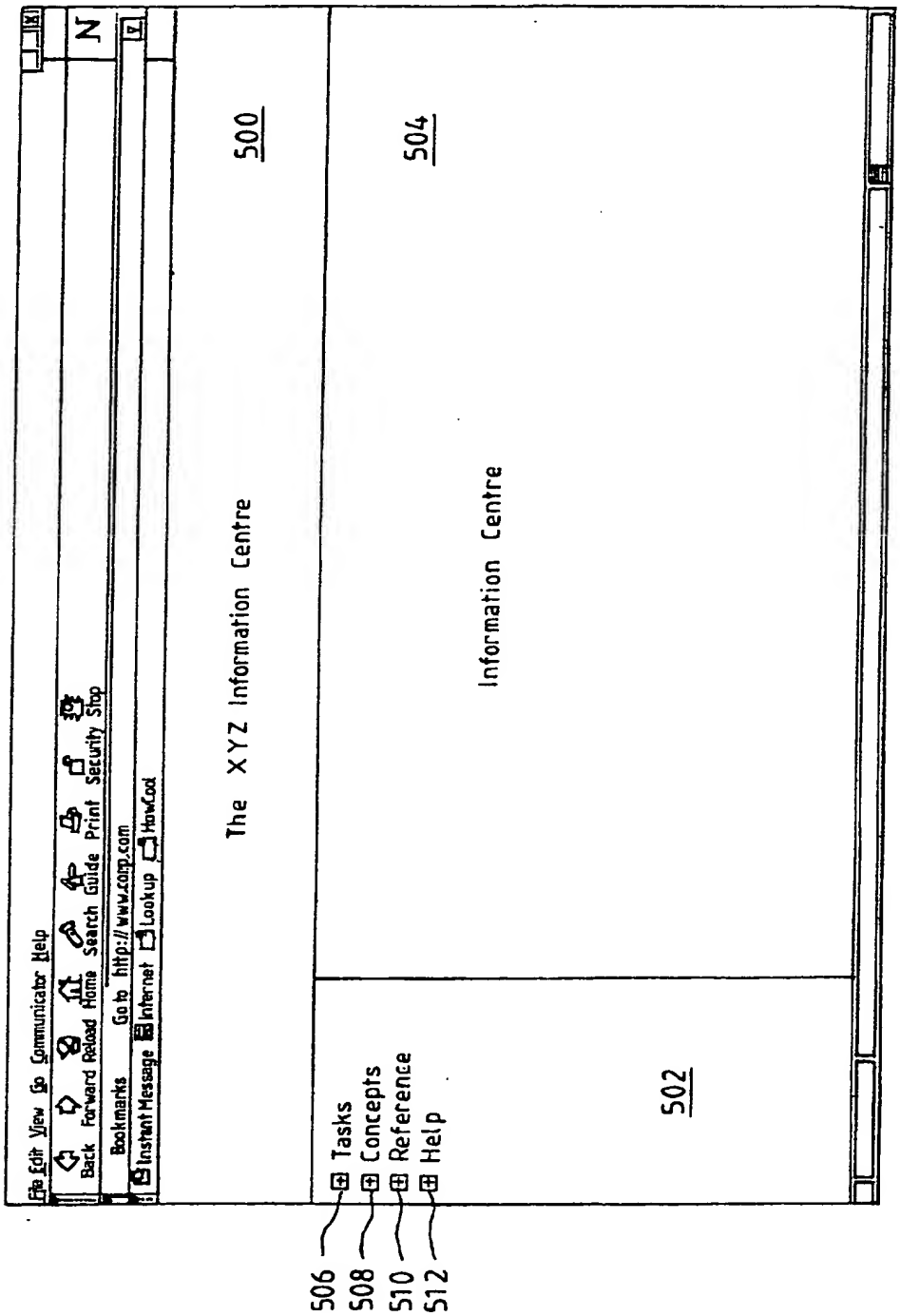


FIG. 5A PRIOR ART

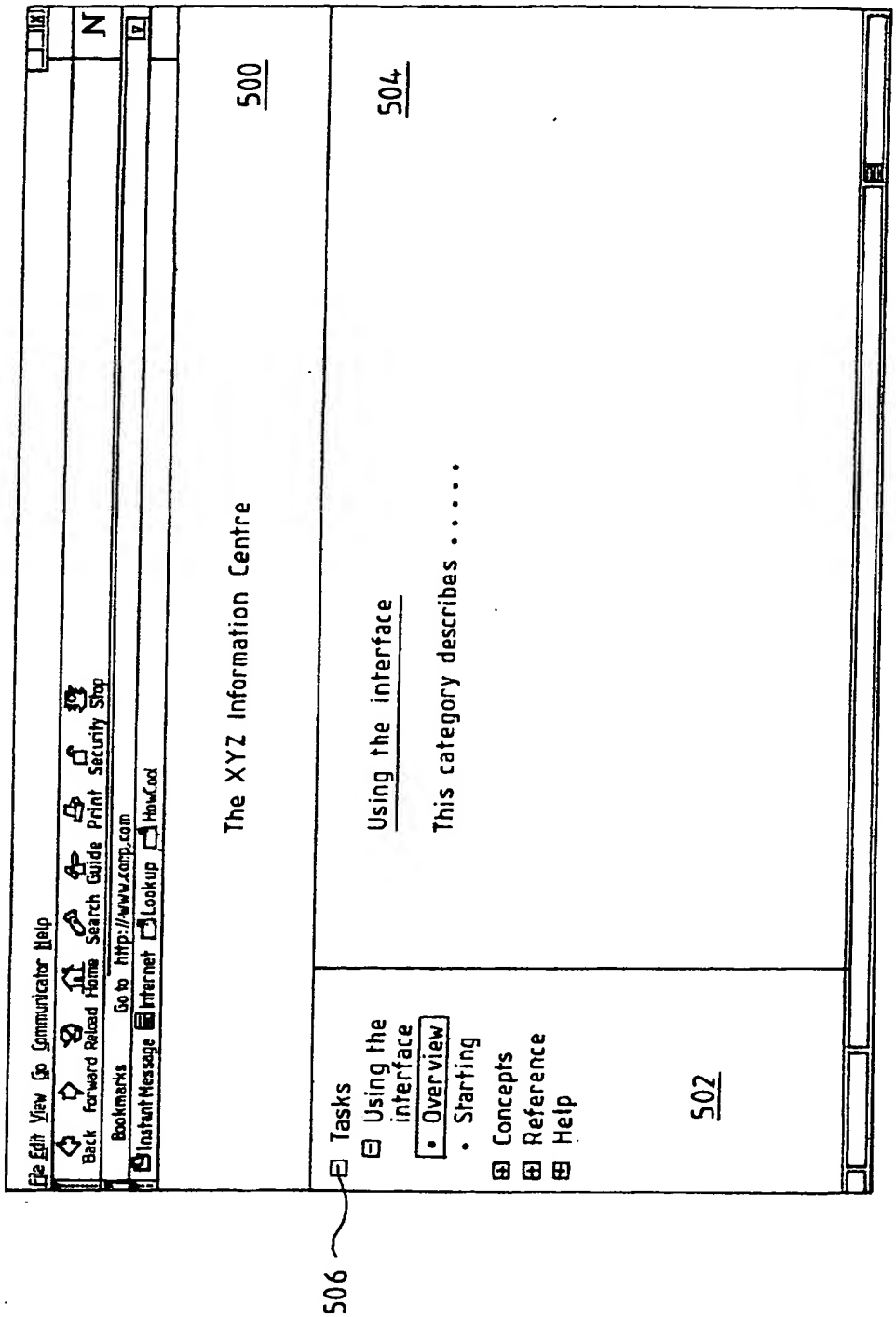


FIG. 5B PRIOR ART

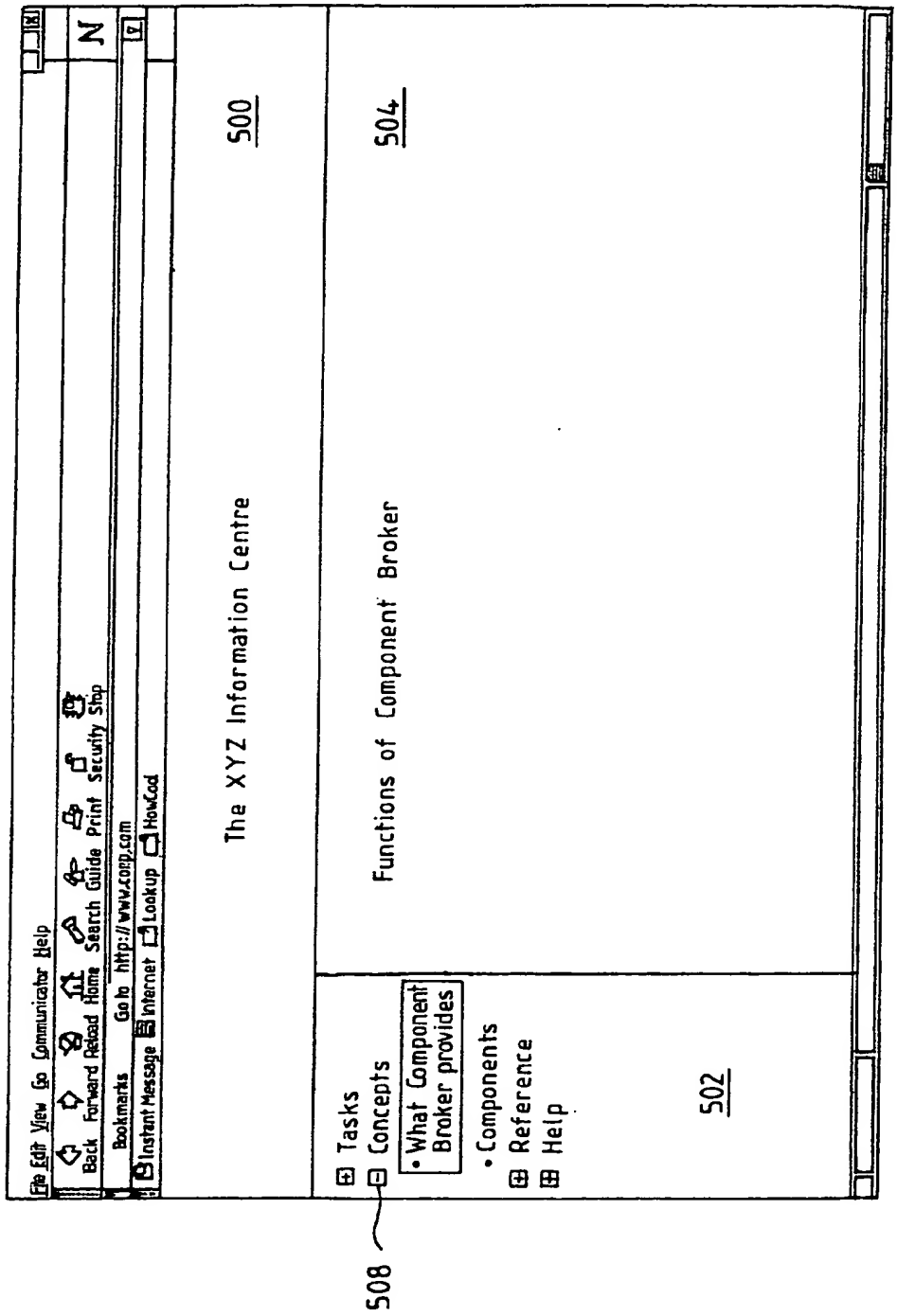


FIG. 5C PRIOR ART

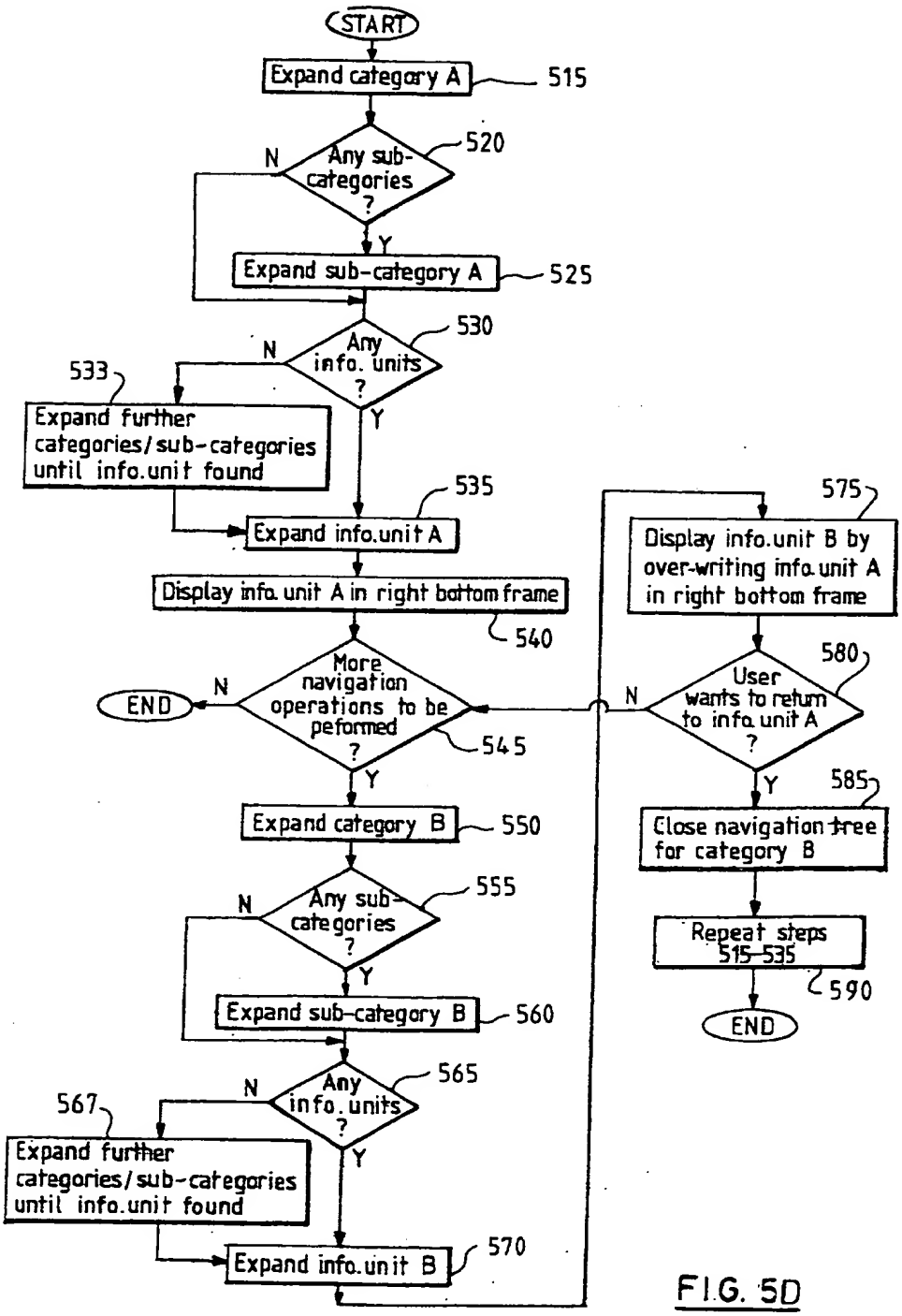

START
Expand category A
515
Any sub-categories ?
520
N
Y
Expand sub-category A
525
Any info. units ?
530
N
Y
533
Expand further categories/sub-categories until info.unit found
Expand info.unit A
535
Display info. unit A in right bottom frame
540
More navigation operations to be peformed ?
545
N
END
Y
Expand category B
550
Any sub-categories ?
555
N
Y
Expand sub-category B
560
Any info. units ?
565
N
Y
567
Expand further categories/sub-categories until info.unit found
Expand info.unit B
570
575
Display info.unit B by over-writing info. unit A in right bottom frame
User wants to return to info. unit A ?
580
N
Y
585
Close navigation tree for category B
Repeat steps 515-535
590
END

FIG. 5D

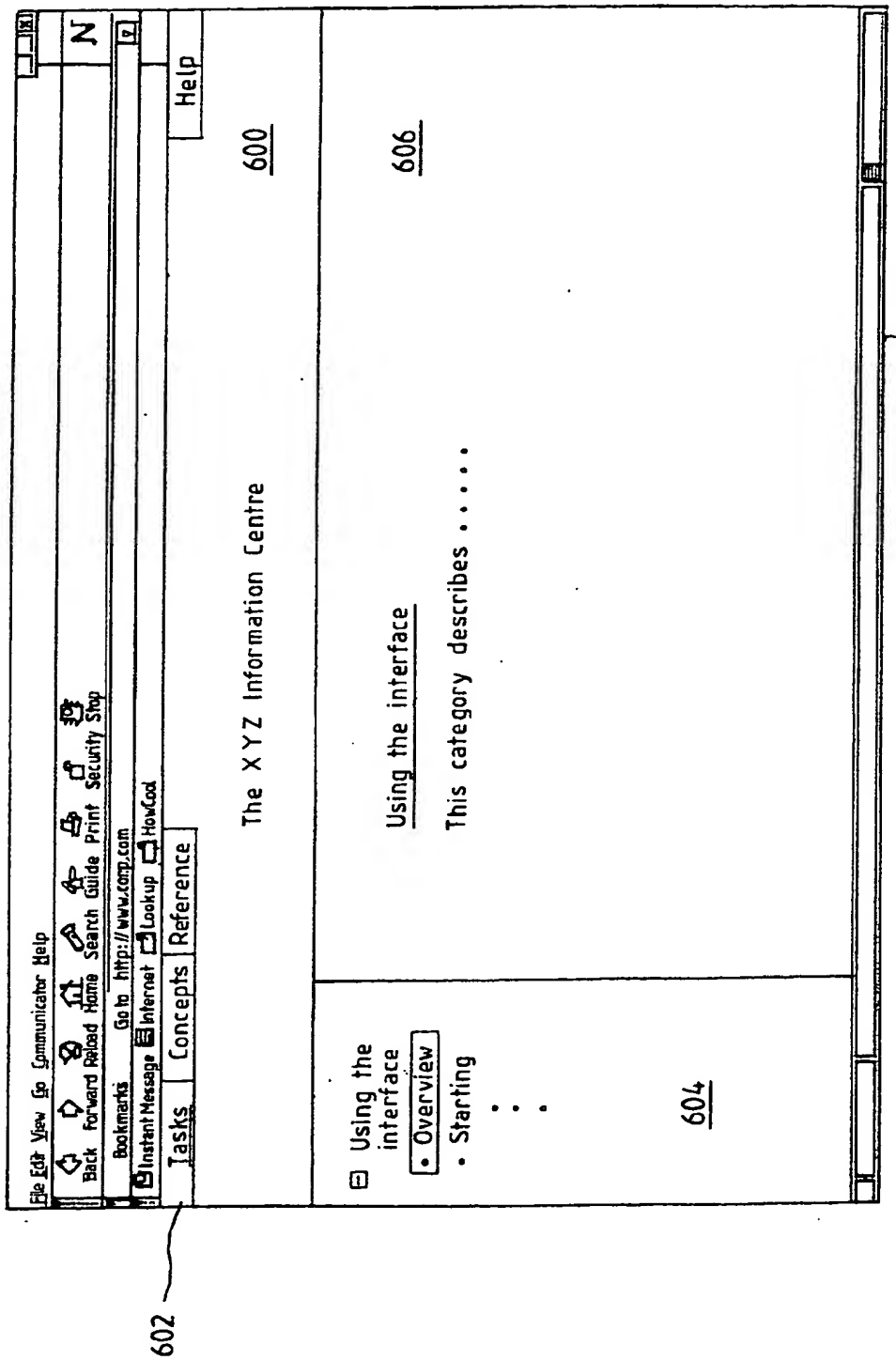

File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Go to http://www.comp.com
Instant Message Internet Lookup HowCool
Tasks
Concepts
Reference
Help
The XYZ Information Centre
600
Using the interface
606
This category describes . . . . .
Using the interface
Overview
Starting
. . .
604
602
610

FIG. 6A

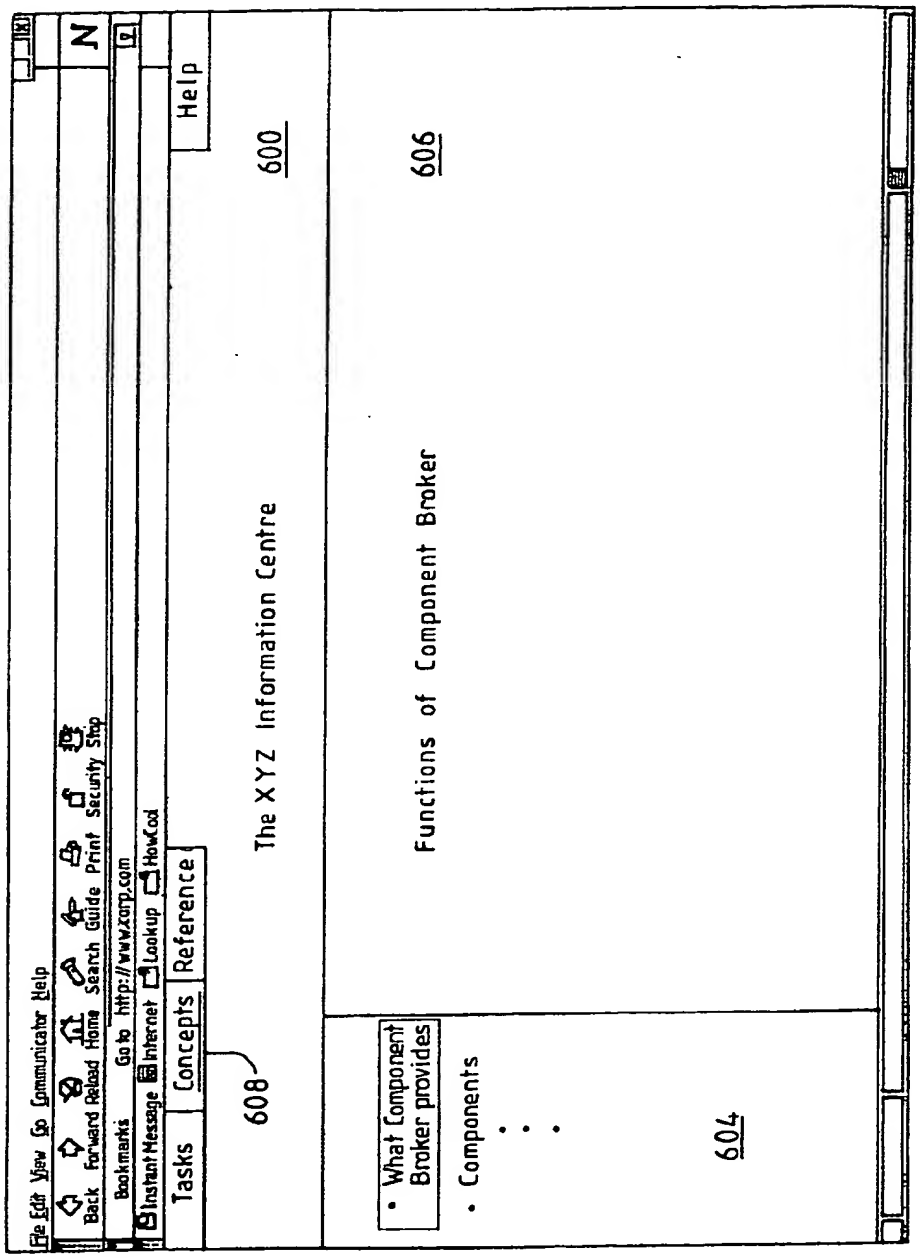

File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Go to http://www.xcorp.com
Instant Message Internet Lookup HowCool
Tasks Concepts Reference Help
608
The XYZ Information Centre
600
• What Component Broker provides
• Components
. . .
604
Functions of Component Broker
606

FIG. 6B

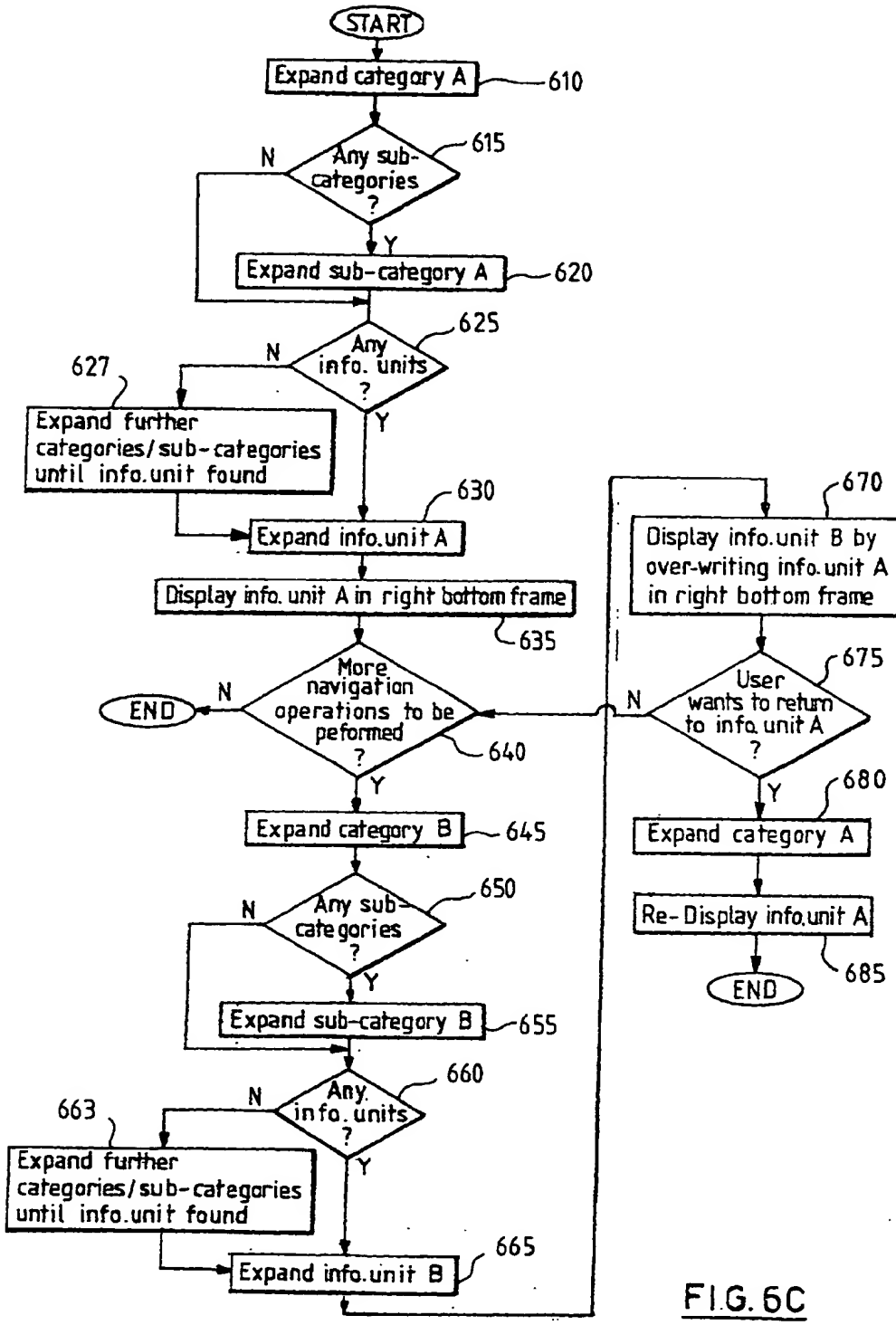

START
Expand category A
610
Any sub-categories ?
615
N
Y
Expand sub-category A
620
Any info. units ?
625
N
627
Expand further categories/sub-categories until info.unit found
Y
Expand info.unit A
630
Display info. unit A in right bottom frame
635
More navigation operations to be peformed ?
640
N
END
Y
Expand category B
645
Any sub-categories ?
650
N
Y
Expand sub-category B
655
Any info. units ?
660
N
663
Expand further categories/sub-categories until info.unit found
Y
Expand info.unit B
665
Display info.unit B by over-writing info.unit A in right bottom frame
670
User wants to return to info. unit A ?
675
N
Y
Expand category A
680
Re-Display info.unit A
685
END

FIG. 6C

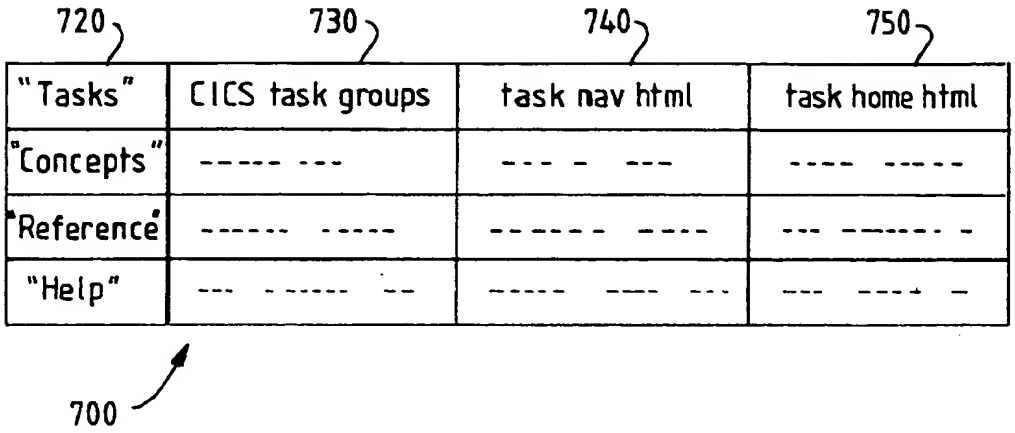

720
730
740
750
"Tasks"
CICS task groups
task nav html
task home html
"Concepts"
"Reference"
"Help"
700

FIG. 7

```
var Category = new Array();                    // Declare the array
                                                  category
var currentCategory = 0;                       // The currently displayed
                                               // category - initially
                                               // the "Tasks" category

function defineCategory (name,desc,navURL,URL) // The array
constructor
{
this.name    = name;        // The name of the category

   this.desc    = desc;     // The description of the category for the
                            // window status area
   this.navURL = navURL;    // The URL of the navigation HTML file
   this.URL     = URL;      // The URL of the introductory information
                               unit
}
// Initialise the array:

Category[0] = new defineCategory ('Tasks',
                                  'CICS task groups',
                                     'tasknav.html',
                                        'taskhome.html');
Category[1] = new defineCategory ('Concepts',
                                  'CICS concepts',
                                     'conceptnav.html',
                                        'concepthome.html');
Category[2] = new defineCategory ('Reference',
                                  'Reference information',
                                     'refnav.html',
                                        'refhome.html');
Category[3] = new defineCategory ('Help',
                                  'Help',
                                     'helpnav.html',
                                        'helphome.html');
```

FIG. 8A

```
// Define the category tabs

for (categoryNumber=0; categoryNumber<Category.length;
categoryNumber++)
{
   tab = '<a id="B' + categoryNumber + '"'
       + ' category="tab"'
       + ' href="' + Category [categoryNumber].navURL + '"'
       + ' target="nav"'
       + '
onMouseOver="window.status=\'+category[categoryNumber].desc+'\';'
                    +'return true;"'
       + ' onClick="switchCategory (categoryNumber);return true;"'
       + '>' + category[categoryNumber].name + '</a>';
   document.writeln(tab);
}
```

FIG. 8B

```
// Switch from the current category to a new category

function switchCategory (newCategory)
{
Category[currentCategory].URL = main.location.href; // Save the
                                                    // currently
                                                    // displayed
                                                    // information
                                                    // unit

   main.location.href = Category[newCategory].URL;  // Redisplay main
                                                    // panel

   switchColours(currentCategory,newCategory);      // Change
                                                    // category tab
                                                    // colours

   currentCategory = newCategory;                   // Remember
                                                    // current
                                                    // category
}
```

FIG. 8C

INFORMATION RETRIEVAL SYSTEM

TECHNICAL FIELD OF THE INVENTION

[0001] The present invention relates to information retrieval in an information network and more particularly, navigating the World Wide Web during a web browsing session.

BACKGROUND OF THE INVENTION

[0002] The World Wide Web is the Internet's multimedia information retrieval system. In the Web environment, client machines communicate with Web servers using the Hypertext Transfer Protocol (HTTP). The web servers provide users with access to files such as text, graphics, images, sound, video, etc., using a standard page description language known as Hypertext Markup Language (HTML). HTML provides basic document formatting and allows the developer to specify connections known as hyperlinks to other servers and files. In the Internet paradigm, a network path to a server is identified by a Uniform Resource Locator (URL) having a special syntax for defining a network connection. So called web browsers, for example, Netscape Navigator (Netscape Navigator is a registered trademark of Netscape Communications Corporation) or Microsoft Internet Explorer, which are applications running on a client machine, enable users to access information by specification of a link via the URL and to navigate between different HTML pages.

[0003] **FIG. 1** shows an example of typical a web browser graphical user interface ("GUI") display on a browser computer. On a portion (**100**) of a user's computer display, a web browser (**110**), in this example Netscape Navigator, runs in its own window. In this example, the web browser is currently pointed to the top-level or home page of the example web site, as indicated by the URL http://www.corp.com in the location bar (**140**). This web page, "index.htm", is configured to split the GUI display into three frames. The upper frame (**120**) is used to display a general banner, the left frame (**130**) is used to display a list of hyperlinks from the top-level web page (or navigation bar), and the right bottom frame (**150**) displays the contents of the currently selected web page. The top of the web browser display includes a row of control icons, including a "back" button (**160**) and a "forward" button (**170**). Additionally, a domain history button (**180**) provides a drop down history list of the URLs of web sites most recently visited by the browser application.

[0004] When the user of the web browser selects a link, the client issues a request to a naming service to map a hostname (in the URL) to a particular network IP (Internet Protocol) address at which the server is located. The naming service returns an IP address that can respond to the request. Using the IP address, the web browser establishes a connection to a server. If the server is available, it returns a web page. To facilitate further navigation within the site, a web page typically includes one or more hypertext references known as "anchors" or "links".

[0005] In **FIG. 2**, a typical tree-like hierarchical organisation of links within a web site is shown, where a given link (**200**) typically points to other Web resources (**210, 215**), and those resources may point to still other resources (**220,222, 224,226,228**). Thus, a given top-level link often has an associated set of lower level links, which may point to still more resources located across many different servers in the network. Navigation through multiple levels of links is often very difficult. The goal of many users of the Internet is to "drill down" to a given piece of information that represents some desired content. Because HTML pages are often statically coded, however, a user often has little choice but to load successive web pages in search of a given web page that might hold the content of interest. This approach is time consuming, and it often results in the user either terminating the navigation of a particular site or simply not finding the relevant content. Additionally, when traversing these multiple levels of links, the user often loses track of the sequence of links used to arrive at a particular web page of interest. Thus, the user could have difficulty in returning to a particular web page after further web pages have been subsequently browsed, if it were necessary to repeat the process of traversing the multiple levels to find that particular web page again.

[0006] A bookmark facility is one way of addressing this problem by providing a mechanism to store and recall specific web pages of interest. Each bookmark comprises the title of the web page and the URL used to access the web page. Additionally, bookmarks often contain the date on which the web page was last visited and the date on which the web page was book-marked, along with additional information.

[0007] Another approach to this navigation problem is provided by built-in navigation functions in currently available web browsers, which use the history log and allow users to revisit previously opened web pages. A "back" button, such as button **160**, backtracks the user's browsing sequence one web page at a time to show the previous web page. After the user has returned to a previous web page, the "forward" button, such as button **170**, is enabled, allowing the user to browse to the forward-most web page in the user's browsing sequence. Sometimes, a user may descend multiple layers into a web site in such a way that the "back" button must be clicked several times to return to the top-level web page.

[0008] While the bookmark and backtracking tools give the user certain limited flexibility in revisiting web pages, the tools limit the user to a single branch in the browsing path. Thus, there is a need for a user to be able to revisit web pages with fewer mouse clicks and also revisit web pages within multiple branches in the browsing path.

[0009] **FIG. 3** shows another approach that enables the user to keep his or her place in a tree of information, that is, the conventional built-in web browser function of opening multiple web browser windows. Therefore, if a particular first web page (**300**) is opened within a first web browser window (**310**) and the user then opens a second web browser window (**330**) to view a second web page (**320**), the first web page (**300**) can be retrieved for viewing by re-selecting the first web browser window (**310**). However, this has the disadvantage of requiring an extra web browser window to be opened which is time-consuming, an extra overhead and also clutters the screen display. Thus, there is a need for a user to be able to keep their place in a tree of information within a single web browser window.

[0010] U.S. patent application Ser. No. 09/310914, filed May 13, 1999 discloses a method and apparatus for implementing direct link selection of cached, previously visited

links in nested web pages. A pointer is added to each web page identifier, which points to the previous linking web page in the navigational path. Another pointer may be added to each web page identifier, which points to the next linking web page in the navigational path. The web browser is thereby enabled to store and display information regarding a navigational path for accessing linking network node addresses.

[0011] U.S. patent application Ser. No. 09/210198, filed Dec. 10, 1998, discloses a recursive link navigation interface method. More specifically, a link map associated with a parent link in a web page is built using a client-side or server-side process. The link map, which includes URLs, text descriptors or actual thumbnail web page images, is selectively displayed at a client web browser when a user takes a given action with respect to the parent link. Thus, for example, when the user hovers over the link with a mouse operation, the link map is displayed to enable the user to determine whether further navigation (through the parent link) is desirable. While the link map is displayed, the user may, alternatively, jump to another link (in the link map) without first traversing the parent and (perhaps other) subordinate links.

[0012] U.S. patent application Ser. No. 09/687091, filed Oct. 10, 2000, discloses a method for browser history thread sibling management. The user may use the conventional "back" and "forward" buttons to traverse backwards and forwards within a browsing history thread, respectively, as well as use the "UP" and "DOWN" keys to traverse to the next and previous sibling browsing history threads, respectively.

[0013] U.S. patent application Ser. No. 09/704596, filed Nov. 3, 2000, discloses a multidimensional browser visual history thread viewer. A user may visually review multiple visual browsing history sessions in a two-dimensional array, or panel, of visual history viewers. The user may select a multidimensional visual history review tool, which causes an array of pop-up viewers to be displayed, each replaying a different web browser visual history session, simultaneously.

[0014] Thus, there is a need to be able to store the web page that is currently being displayed, so that when a user chooses another navigation path and arrives at a different web page, it is possible to switch between the web pages contained within different navigation paths, with ease.

SUMMARY OF THE INVENTION

[0015] Accordingly, in a first aspect, the present invention provides a method of retrieving information by navigating within a web browsing session, in which the information is stored in a hierarchical tree comprising a root node, a plurality of top-level child nodes representing information categories, and a plurality of leaf nodes, each of said nodes having an associated information unit, said method comprising the steps of: displaying an associated information unit of the root node in an information space; performing navigation operations from the root node, by selecting a first of said plurality of top-level child nodes, traversing said plurality of leaf nodes and selecting a first of said plurality of leaf nodes; in response to said performing step, displaying a first associated information unit of said first of said plurality of leaf nodes in said information space; storing said first associated information unit; repeating each of said performing step, said displaying step and said storing step, for a second of said plurality of top-level child nodes whereby a second associated information unit is displayed in said information space; re-selecting said first of said plurality of top-level child nodes, and in response to said re-selecting step, automatically re-displaying said first associated information unit in said information space.

[0016] Preferably, the nodes are selected by clicking on an associated icon. Preferably, the first associated information unit is re-displayed by clicking once on an associated icon of the first of said plurality of top-level child nodes. Preferably, a selected associated icon changes colour and presentation characteristics of the associated information units are obtained from a style sheet.

[0017] In a preferred embodiment of the present invention, the root node and each of the plurality of top-level child nodes has an associated navigation tree, whereby the navigation tree has information links that are expandable from a closed to an expanded state. Preferably, the state of a navigation tree is stored upon closing of the navigation tree and is restored upon opening of the navigation tree. Preferably, the state of a navigation tree is stored on a client computer.

[0018] Preferably, data associated with the root node and each of the plurality of top-level child nodes is stored as items in an array. For one of the nodes, the data comprises: text associated with the one node; a network address associated with a navigation tree for the one node, and a network address associated with a first associated information unit returned by traversing through the one node.

[0019] Preferably, the information is organised as an information center of a collection of online documents, whereby the information center comprises an upper frame containing the associated icons, a left frame containing a navigation tree and a right bottom frame containing the information space. Preferably, the information center is structured in two dimensions.

[0020] In a second aspect, the present invention provides a computer program comprising instructions which when executed on a computer cause said computer, in response to user inputs, to carry out the method as described above.

[0021] In a third aspect, the present invention provides a client computer for retrieving information by navigating within a web browsing session, whereby said client is connected via a network to a server in which the information is stored in a hierarchical tree comprising a root node, a plurality of top-level child nodes representing information categories, and a plurality of leaf nodes, each of said nodes having an associated information unit, said client computer comprising: means for displaying an associated information unit of the root node in an information space; means for performing navigation operations from the root node, by selecting one of said plurality of top-level child nodes, traversing said plurality of leaf nodes and selecting one of said plurality of leaf nodes; means, responsive to selection of said one of said plurality of leaf nodes, for displaying an associated information unit of said one of said plurality of leaf nodes in said information space; means for storing said associated information unit; means, responsive to re-selection of said one of said plurality of top-level child nodes, for

automatically re-displaying said associated information unit in said information space, notwithstanding the intervening selection by said means for performing navigation of a different one of said plurality of top-level child nodes.

[0022] In a fourth aspect, the present invention provides a system for retrieving information by navigating within a web browsing session, comprising: a client as described above and a server having means for storing the information in a hierarchical tree comprising a root node, a plurality of top-level child nodes representing information categories, and a plurality of leaf nodes, each of said nodes having an associated information unit, whereby said server is responsive to receiving inputs from the client to provide an associated information unit on selection of one of said nodes.

BRIEF DESCRIPTION OF THE DRAWINGS

[0023] The present invention will now be described, by way of example only, with reference to preferred embodiments thereof, as illustrated in the following drawings:

[0024] **FIG. 1** shows a prior art example of display in a typical web browser graphical interface window;

[0025] **FIG. 2** shows a prior art tree structure of hyperlink relationships;

[0026] **FIG. 3** shows a prior art technique for displaying multiple web browser windows;

[0027] **FIG. 4** shows a prior art distributed data processing system in which the present invention may be implemented;

[0028] **FIG. 5A** shows a prior art information center;

[0029] **FIG. 5B** shows the results of navigation operations performed on the information center of **FIG. 5A**;

[0030] **FIG. 5C** shows the results of further navigation operations performed on the information center of **FIG. 5A**;

[0031] **FIG. 5D** is a flow chart showing the operational steps involved in performing the navigation operations of **FIGS. 5B and 5C**;

[0032] **FIG. 6A** shows the results of navigation operations performed on an information center, according to the present invention;

[0033] **FIG. 6B** shows the results of further navigation operations performed on an information center, according to the present invention;

[0034] **FIG. 6C** is a flow chart showing the operational steps involved in performing the navigation operations of **FIGS. 6A and 6B**, according to the present invention;

[0035] **FIG. 7** shows a representation of an array, whereby information related to a top-level category is stored as items in the array, according to the present invention;

[0036] **FIG. 8A** shows an example of the declaration and initialisation of the array, according to the present invention;

[0037] **FIG. 8B** shows an example of the definition of top-level category tabs, according to the present invention, and **FIG. 8C** shows an example of a function, which is invoked when a user switches between top-level categories.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

[0038] **FIG. 4** shows a pictorial representation of a distributed data processing system in which the present invention may be implemented. Distributed data processing system **400** comprises a number of computers, connected by a network **402**. Server **404** is connected to network **402** along with storage unit **406** and clients **408, 410** and **412**. In the depicted example, distributed data processing system **400** is the Internet, with network **402** representing a world-wide collection of networks and gateways that use the transmission control protocol over internet protocol (TCP/IP) suite of protocols to communicate with one another.

[0039] In a preferred embodiment of the present invention, an improved information center is described. Information centres are online collections of documents based on HTML web browsers. The information centres provide easy access to large libraries of softcopy information about particular products, such as help and support information, via a simple interface.

[0040] **FIG. 5A** shows a typical interface of a current information center, which is supported by a web browser as shown in **FIG. 1**. The upper frame **(500)** contains a title and other high-level items of information and links. Below the upper frame **(500)**, the left frame **(502)** contains a navigation bar to display links to information units. Once a link is selected, the appropriate information unit is displayed in the right bottom frame **(504)**. In this example, in order to assist users to find information easily, the information units are divided into various categories. Referring to the navigation bar **(502)**, the category "Tasks" contains information units describing how to perform specific activities. The category "Concepts" contains information units detailing background information of product features and the category "Reference" contains information units with tabular or diagrammatic information.

[0041] When using an information center such as the one in **FIG. 5A**, only a single information unit can be displayed at a time in the right bottom frame **(504)**. If another new information unit is to be displayed, the first is discarded and is replaced completely by the new information unit. Frequently, it will be necessary to navigate to the new information unit using the navigation tree in the navigation bar **(502)**, whereby the navigation tree can be opened to a given level and closed to a given level. Therefore, several mouse clicks or the equivalent may be required to move from an information unit in one category to an information unit in another category. Subsequently, if the user wishes to return to the first information unit, several further mouse clicks may be required to re-establish the navigation tree. Alternatively navigation could occur by using the mouse to click on hyperlinks displayed in the right bottom frame **(504)**. Also, in **FIG. 5A**, to the left of each category, an icon **(506, 508, 510, 512)** containing a square is displayed, to indicate that further sub-categories are contained within the top-level category. It should be understood that the term icon also covers graphic symbols such as tabs and buttons.

[0042] Referring to **FIGS. 5B and 5C**, which are used in conjunction with the flow chart of **FIG. 5D**, once a user clicks on the icon **(506)** containing a square to expand (step **515**) the category "Tasks", further sub-categories are displayed. Subsequently the user clicks to expand (step **525**)

one of these sub-categories, namely, "Using the Interface" and then clicks to expand (step **535**) an information unit, namely, "Overview", contained in this sub-category. The information unit is now displayed (step **540**) in the right bottom frame (**504**). If there are no sub-categories for selection, under the top-level category (step **520**), the user may click directly on an information unit. However, if there are no information units for selection (step **530**), the user has the option to click (step **533**) either on a different top-level or sub-category, in order to continue with the navigation operations.

[**0043**] Supposing the user now wants to display an information unit contained within another top-level category by performing (step **545**) further navigation operations. In **FIG. 5C**, the user expands (step **550**) the "Concepts" category by clicking on the icon (**508**) containing a square and this operation causes the navigation tree under category "Tasks" to be closed. It is now possible to navigate to an information unit under category "Concepts", for example, by directly clicking (step **570**) on the "What Component Broker provides" information unit. This is the consequence of a negative result to step **555** and a positive result to step **565**.

[**0044**] As in **FIG. 5C**, if there are no sub-categories for selection under the top-level category (step **555**), the user may click directly (step **570**) on an information unit. However, if there are no information units for selection (step **565**), the user has the option to click (step **567**) either on a different top-level or sub-category, in order to continue with the navigation operations.

[**0045**] The information unit "What Component Broker provides", is now displayed (step **575**) by replacing the "Overview" information unit in the right bottom frame (**504**). If the user does not want to return to the original "Overview" information unit (step **580**), and no further navigation operations are to be performed (step **545**), the navigation process ends.

[**0046**] However, if the user wants to return to the original "Overview" information unit contained within "Tasks"; "Using the Interface"; (step **580**), the user is required to first close (step **585**) the "Concepts" navigation tree. The user then re-expands the "Tasks" navigation tree and thirdly navigates down the "Tasks" navigation tree once more (step **590**). Obviously, this is a tedious and time-consuming process, which can be frustrating for the user.

[**0047**] In a preferred embodiment of the present invention, the navigation process is simplified by implementing a two-dimensional navigation structure rather than a one-dimensional navigation structure. That is, each category of information has its own navigation tree and selection of categories is moved from a one-dimensional navigation bar to a second dimension, namely, via tabs in the upper frame. Each category of information has an associated tab. Alternatively a three-dimensional navigation structure could be implemented although this structure may be an overhead and may also be too complex.

[**0048**] Additionally, when a user closes a top-level category, the state of the sub-categories within the navigation tree is stored, that is, which sub-categories are in an expanded state and which are in a closed state. The state information is stored on the client machine so that when the top-level category is re-expanded, the navigation tree for that top-level category is restored to its previous state and the information unit that was last displayed is also restored. This is advantageous, in that it is now possible to switch between information units held within different categories with only a single mouse click.

[**0049**] As an example, referring to **FIGS. 6A and 6B**, to be used in conjunction with **FIG. 6C**, a user first clicks (step **610**) on the tab (**602**) for category "Tasks", in the upper frame (**600**). The user then navigates down the navigation tree for "Tasks", via steps **615** through **630**, to an information unit, namely, "Overview", which is displayed (step **635**) in the right bottom frame (**606**). If further navigation operations are to be performed (step **640**), in **FIG. 6B**, the tab (**608**) for category "Concepts" is clicked (step **645**) on in the upper frame (**600**), for example. The user navigates down this tree, via steps **650** through **665**, to an information unit, namely, "What Component Broker provides". This is displayed (step **670**) by replacing the "Overview" information unit in the right bottom frame (**606**). If the user does not want to return to the original "Overview" information unit (step **675**), and no further navigation operations are to be performed (step **640**), the navigation process ends.

[**0050**] In the preferred embodiment of the present invention, if the user wishes to return to information unit "Overview" (step **675**), the user only has to use a single mouse click on the tab (**602**) for category "Tasks", in order to re-select (step **680**) that category. The information unit as shown in **FIG. 6A** is then re-displayed automatically (step **685**).

[**0051**] The preferred embodiment of the present invention is implemented in the JavaScript programming language. (JavaScript is a trademark of Sun Microsystems Inc.) Specifically, information associated with each category is held as an item in a JavaScript array. A representation of the array (**700**) is shown in **FIG. 7**. An example of the array declaration and initialisation is shown in **FIG. 8A**.

[**0052**] For example, taking the category "Tasks", the following information is initially held in the array (**700**):

[**0053**] 1. Text (**720**) associated with the category tabs in the upper frame (**600**) and text (**730**) to be displayed in a window status area (**610**) or a pop up window for example, when a mouse cursor is positioned over this category tab

[**0054**] 2. A URL (**740**) associated with the HTML file used to build the navigation tree for "Tasks", whereby the navigation tree is displayed in the navigation bar (**604**)

[**0055**] 3. A URL (**750**) associated with the information unit last displayed in the right bottom frame (**606**), whereby the information unit is contained within the category "Tasks".

[**0056**] The array (**700**) would therefore initially contain the following information for category "Tasks":

[**0057**] "Tasks"; "CICS task groups"; tasknav.html; taskhome.html;

[**0058**] In **FIG. 8B**, the tabs for the information categories are defined. Preferably, the id=attribute is used by a function that changes the colours of a category tab, depending on whether the tab is currently selected.

[0059] In this case, the switchColors function, as shown in FIG. 8C, uses the id=attribute. In FIG. 8B, the category="tab" attribute is used to obtain presentation information from a cascading style sheet. Style sheets are programmatic representations of the processing operations or transformations to be performed on input data to create an output with desired presentation characteristics. For example, presentation characteristics such as the size, weight and name of the font to be used can be described. In FIG. 8B, referring to the target="nav" attribute, "nav" is the JavaScript name of the left frame (604), which contains the navigation bar.

[0060] When a user wishes to switch from one category of information, for example "Tasks", to a new category, for example "Reference", he or she does so either by clicking on the "Reference" tab, or by using the "Tab" key to move to the "Reference" tab and then pressing the "Enter" key. In both cases, as shown in FIG. 8B, the onClick event handler associated with the tab, is invoked by the web browser. This in turn invokes the JavaScript function switchCategory, defined in FIG. 8C, and passes it an associated parameter. In this case the parameter is the index (categoryNumber) of the "Reference" category into the array of category definitions, namely, 2.

[0061] The switchCategory function first stores the URL associated with the currently displayed information unit. The URL is stored in the array item corresponding to the current category, in this example, "Tasks". The switchCategory function then sets the right bottom frame (606) to display the new information unit identified by the URL, that was either initially stored in the array (700) or was stored on a previous invocation of the switchCategory function on leaving the category "Reference". Optionally, the switchCategory function may then indicate which category the user is currently viewing, by changing the colours of the previous and current tabs. Finally the switchCategory function stores the index of the new category, in this case "Reference", as the current category.

[0062] In FIG. 8B, on return from the function switchCategory, the web browser uses the values of the href=and target=attributes from the anchor tag of the new category to invoke the HTML file that builds the navigation tree in the navigation bar (604), for that category.

[0063] The present invention could also be applied to web sites other than to a web site for an information center containing detailed product information. For example, in a corporate web site, category "Products" would contain information units describing various product lines, category "Services" would describe the consultancy services offered, category "Support" would describe any after-sales services and category "About the company" would provide general information.

1. A method of retrieving information by navigating within a web browsing session, in which the information is stored in a hierarchical tree comprising a root node, a plurality of top-level child nodes representing information categories, and a plurality of leaf nodes, each of said nodes having an associated information unit, said method comprising the steps of:

displaying an associated information unit of the root node in an information space;

performing navigation operations from the root node, by selecting a first of said plurality of top-level child nodes, traversing said plurality of leaf nodes and selecting a first of said plurality of leaf nodes;

in response to said performing step, displaying a first associated information unit of said first of said plurality of leaf nodes in said information space;

storing said first associated information unit;

repeating each of said performing step, said displaying step and said storing step, for a second of said plurality of top-level child nodes whereby a second associated information unit is displayed in said information space;

re-selecting said first of said plurality of top-level child nodes, and

in response to said re-selecting step, automatically re-displaying said first associated information unit in said information space.

2. A method as claimed in claim 1, wherein said nodes are selected by clicking on an associated icon.

3. A method as claimed in claim 2, wherein said first associated information unit is re-displayed by clicking once on an associated icon of said first of said plurality of top-level child nodes.

4. A method as claimed in claim 1, wherein said root node and each of said plurality of top-level child nodes has an associated navigation tree, whereby said navigation tree has information links that are expandable from a closed to an expanded state.

5. A method as claimed in claim 4, wherein the state of a navigation tree is stored upon closing of said navigation tree and the state of said navigation tree is restored upon opening of said navigation tree.

6. A method as claimed in claim 4, wherein the state of a navigation tree is stored on a client computer.

7. A method as claimed in claim 1, wherein data associated with said root node and each of said plurality of top-level child nodes is stored as items in an array.

8. A method as claimed in claim 7, wherein, for one of said nodes, said data comprises: text associated with said one of said nodes; a network address associated with a navigation tree for said one of said nodes, and a network address associated with a first associated information unit returned by traversing through said one of said nodes.

9. A method as claimed in claim 1, wherein said information is organised as an information center of a collection of online documents, whereby said information center comprises an upper frame containing associated icons, a left frame containing a navigation tree and a right bottom frame containing said information space.

10. A method as claimed in claim 9, wherein said information center is structured in two dimensions.

11. A method as claimed in claim 2, wherein a selected associated icon changes colour.

12. A method as claimed in claim 1, wherein presentation characteristics of an associated information unit is obtained from a style sheet.

13. A computer program product stored on a computer readable storage medium comprising instructions which when executed on a computer cause said computer, in response to user inputs, to carry out the method as claimed in claim 1.

14. A client computer for retrieving information by navigating within a web browsing session, whereby said client is connected via a network to a server in which the information is stored in a hierarchical tree comprising a root node, a plurality of top-level child nodes representing information categories, and a plurality of leaf nodes, each of said nodes having an associated information unit, said client computer comprising:

means for displaying an associated information unit of the root node in an information space;

means for performing navigation operations from the root node, by selecting one of said plurality of top-level child nodes, traversing said plurality of leaf nodes and selecting one of said plurality of leaf nodes;

means, responsive to selection of said one of said plurality of leaf nodes, for displaying an associated information unit of said one of said plurality of leaf nodes in said information space;

means for storing said associated information unit;

means, responsive to re-selection of said one of said plurality of top-level child nodes, for automatically re-displaying said associated information unit in said information space, notwithstanding the intervening selection by said means for performing navigation of a different one of said plurality of top-level child nodes.

15. A client computer as claimed in claim 14, wherein said nodes are selected by clicking on an associated icon.

16. A client computer as claimed in claim 15, wherein said first associated information unit is re-displayed by clicking once on an associated icon of said first of said plurality of top-level child nodes.

17. A client computer as claimed in claim 14, wherein said root node and each of said plurality of top-level child nodes has an associated navigation tree, whereby said navigation tree has information links that are expandable from a closed to an expanded state.

18. A client computer as claimed in claim 17, wherein the state of a navigation tree is stored upon closing of said navigation tree and the state of said navigation tree is restored upon opening of said navigation tree.

19. A client computer as claimed in claim 14, wherein data associated with said root node and each of said plurality of top-level child nodes is stored as items in an array.

20. A client computer as claimed in claim 19, wherein, for one of said nodes, said data comprises: text associated with said one of said nodes; a network address associated with a navigation tree for said one of said nodes, and a network address associated with a first associated information unit returned by traversing through said one of said nodes.

21. A client computer as claimed in claim 14, wherein said information is organised as an information center of a collection of online documents, whereby said information center comprises an upper frame containing associated icons, a left frame containing a navigation tree and a right bottom frame containing said information space.

22. A client computer as claimed in claim 21, wherein said information center is structured in two dimensions.

23. A client computer as claimed in claim 15, wherein a selected associated icon changes colour.

24. A client computer as claimed in claim 14, wherein presentation characteristics of an associated information unit is obtained from a style sheet.

25. A system for retrieving information by navigating within a web browsing session, comprising:

a server having means for storing the information in a hierarchical tree comprising a root node, a plurality of top-level child nodes representing information categories, and a plurality of leaf nodes, each of said nodes having an associated information unit, whereby said server is responsive to receiving inputs from the client to provide an associated information unit on selection of one of said nodes, and

a client computer for retrieving information by navigating within a web browsing session, whereby said client is connected via a network to said server, said client computer comprising:

means for displaying an associated information unit of the root node in an information space;

means for performing navigation operations from the root node, by selecting one of said plurality of top-level child nodes, traversing said plurality of leaf nodes and selecting one of said plurality of leaf nodes;

means, responsive to selection of said one of said plurality of leaf nodes, for displaying an associated information unit of said one of said plurality of leaf nodes in said information space;

means for storing said associated information unit;

means, responsive to re-selection of said one of said plurality of top-level child nodes, for automatically re-displaying said associated information unit in said information space, notwithstanding the intervening selection by said means for performing navigation of a different one of said plurality of top-level child nodes.

26. A system as claimed in claim 25, wherein said nodes are selected by clicking on an associated icon.

27. A system as claimed in claim 26, wherein said first associated information unit is re-displayed by clicking once on an associated icon of said first of said plurality of top-level child nodes.

28. A system as claimed in claim 25, wherein said root node and each of said plurality of top-level child nodes has an associated navigation tree, whereby said navigation tree has information links that are expandable from a closed to an expanded state.

29. A system as claimed in claim 28, wherein the state of a navigation tree is stored upon closing of said navigation tree and the state of said navigation tree is restored upon opening of said navigation tree.

30. A system as claimed in claim 28, wherein the state of a navigation tree is stored on said client computer.

31. A system as claimed in claim 25, wherein data associated with said root node and each of said plurality of top-level child nodes is stored as items in an array.

32. A system as claimed in claim 31, wherein, for one of said nodes, said data comprises: text associated with said one of said nodes; a network address associated with a navigation tree for said one of said nodes, and a network address associated with a first associated information unit returned by traversing through said one of said nodes.

33. A system as claimed in claim 25, wherein said information is organised as an information center of a

collection of online documents, whereby said information center comprises an upper frame containing associated icons, a left frame containing a navigation tree and a right bottom frame containing said information space.

34. A system as claimed in claim 33, wherein said information center is structured in two dimensions.

35. A system as claimed in claim 26, wherein a selected associated icon changes colour. 36. A system as claimed in claim 25, wherein presentation characteristics of an associated information unit is obtained from a style sheet.

* * * * *

US 20020010709A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2002/0010709 A1**

Culbert et al. (43) **Pub. Date: Jan. 24, 2002**

(54) **METHOD AND SYSTEM FOR DISTILLING CONTENT**

(76) Inventors: **Daniel Jason Culbert**, Sunnyvale, CA (US); **Denis Gulsen**, Redwood City, CA (US)

Correspondence Address:
Daniel Culbert
1035 Aster Ave # 2196
Sunnyvale, CA 94086 (US)

(21) Appl. No.: **09/792,522**

(22) Filed: **Feb. 26, 2001**

Related U.S. Application Data

(63) Non-provisional of provisional application No. 60/184,068, filed on Feb. 22, 2000.

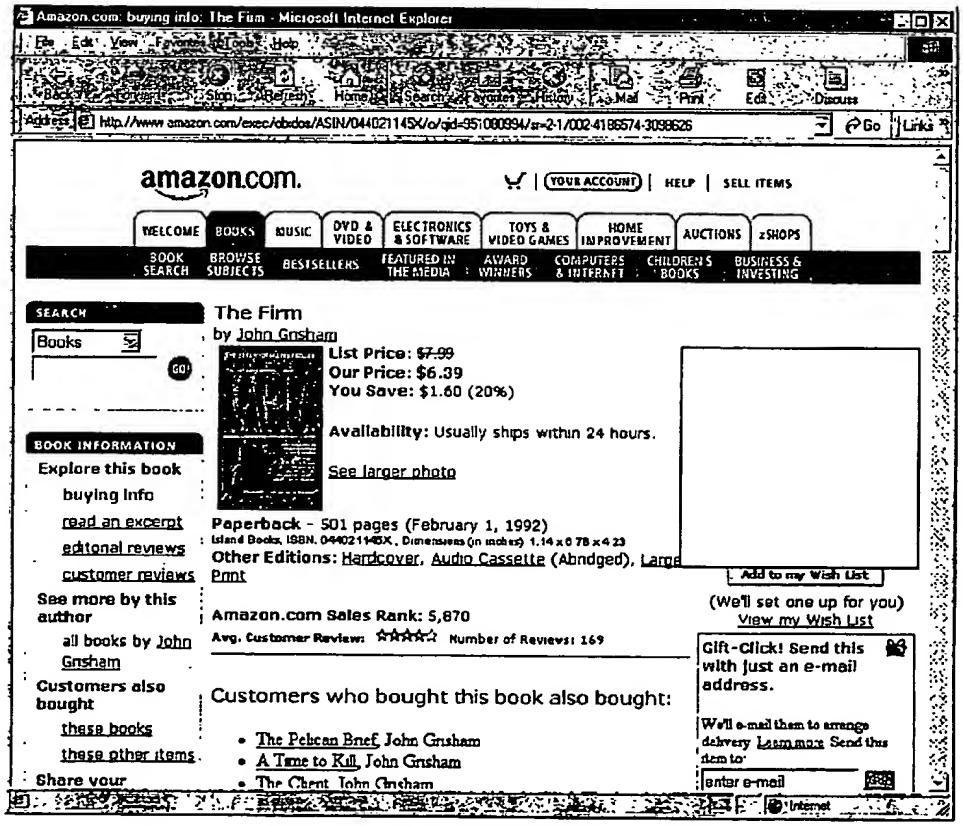
Publication Classification

(51) **Int. Cl.**[7] **G06F 17/00**

(52) **U.S. Cl.** **707/500**; 709/1

(57) **ABSTRACT**

This is a system and method for processing and selectively storing content of an Internet web site. A key aspect of each variation of the invention is the distillation of information associated with an Internet location to which the user has browsed using various algorithms operating in the background to produce a linked group of distilled pieces of information (a "datagram") which may be used in various ways for or by the user.



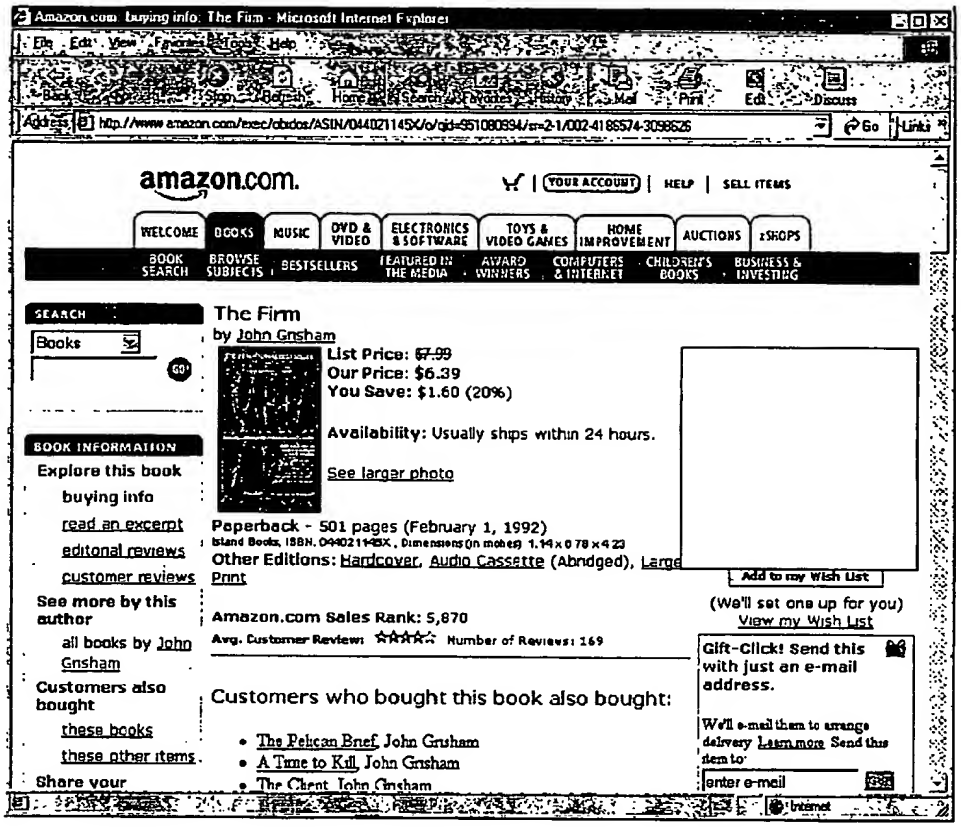

Amazon.com: Buying Info: The Firm - Microsoft Internet Explorer
amazon.com.
YOUR ACCOUNT | HELP | SELL ITEMS
WELCOME BOOKS MUSIC DVD & VIDEO ELECTRONICS & SOFTWARE TOYS & VIDEO GAMES HOME IMPROVEMENT AUCTIONS zSHOPS
BOOK SEARCH BROWSE SUBJECTS BESTSELLERS FEATURED IN THE MEDIA AWARD WINNERS COMPUTERS & INTERNET CHILDREN'S BOOKS BUSINESS & INVESTING
SEARCH
Books
GO
BOOK INFORMATION
Explore this book
buying info
read an excerpt
editorial reviews
customer reviews
See more by this author
all books by John Grisham
Customers also bought
these books
these other items
Share your
The Firm
by John Grisham
List Price: $7.99
Our Price: $6.39
You Save: $1.60 (20%)
Availability: Usually ships within 24 hours.
See larger photo
Paperback - 501 pages (February 1, 1992)
Other Editions: Hardcover, Audio Cassette (Abridged), Large Print
Amazon.com Sales Rank: 5,870
Avg. Customer Review: Number of Reviews: 169
Add to my Wish List
(We'll set one up for you)
View my Wish List
Gift-Click! Send this with just an e-mail address.
We'll e-mail them to arrange delivery Learn more Send this item to:
enter e-mail
Customers who bought this book also bought:
The Pelican Brief, John Grisham
A Time to Kill, John Grisham
The Client, John Grisham
Internet

Figure 1.

```
<HTML>
<HEAD>
<TITLE>Amazon.com: buying info: The Firm</TITLE>
</HEAD>
<body bgcolor="#FFFFFF" link="#003399" alink="#FF9933"
vlink="#996633" text="#000000">
<a name="top"><!--Top of Page--></a>
<table border=0 width=100% cellspacing=0 cellpadding=0>
<tr align=center bgcolor="#FFFFFF">
<td><a href=/g/v11/nav/top-nav.map/002-4186574-3098626 ><img width=590
height=75 border=0 src="/g/post-holiday/logo-2000-nav/product-type-
books.gif" alt="Welcome to Amazon.com's Book Store" USEMAP="#top_nav_map"
ISMAP></a></td>
</tr>
<tr align=center bgcolor="#336633">
<td><a href=/g/v10/nav/books-nav.map/002-4186574-3098626 ><img width=590
height=28 border=0 src="/g/v10/nav/books-nav-default.gif" alt=""
USEMAP="#books_nav_map" ISMAP></a></td>
</tr>
</table>
<map name="top_nav_map">
...
</map>
<map name="books_nav_map">
...
<area shape="rect" coords="514,0,589,27" href=/exec/obidos/ts/browse-
books/3/ref=b_tn_bu/002-4186574-3098626>
</map>
...
<tr><td> </td><td width=18> </td><td bgcolor="#FFFFFF"><font
face=verdana,arial,helvetica size=-1 color=#CC6600><b>buying
info</b></font></td></tr>
```

Figure 2.

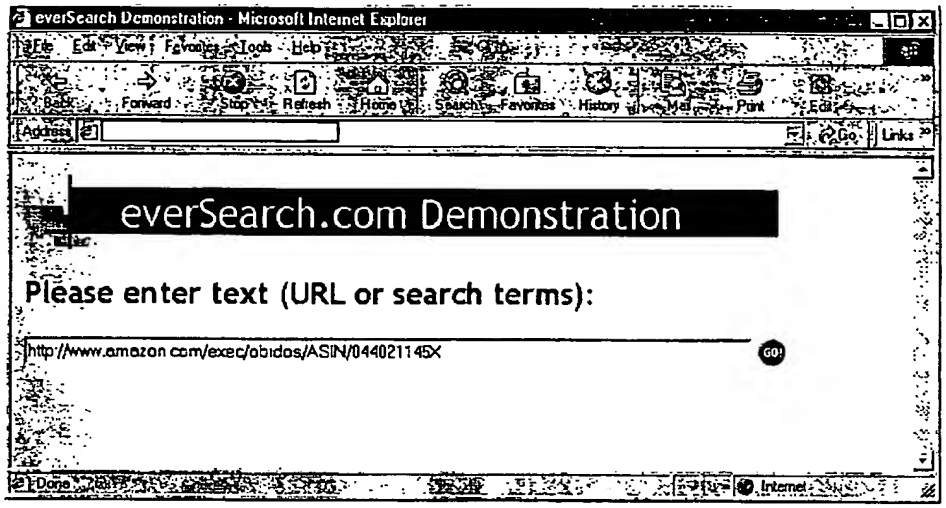

everSearch Demonstration - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address
Go
Links
everSearch.com Demonstration
Please enter text (URL or search terms):
http://www.amazon.com/exec/obidos/ASIN/044021145X
GO!
Done
Internet

Figure 3.

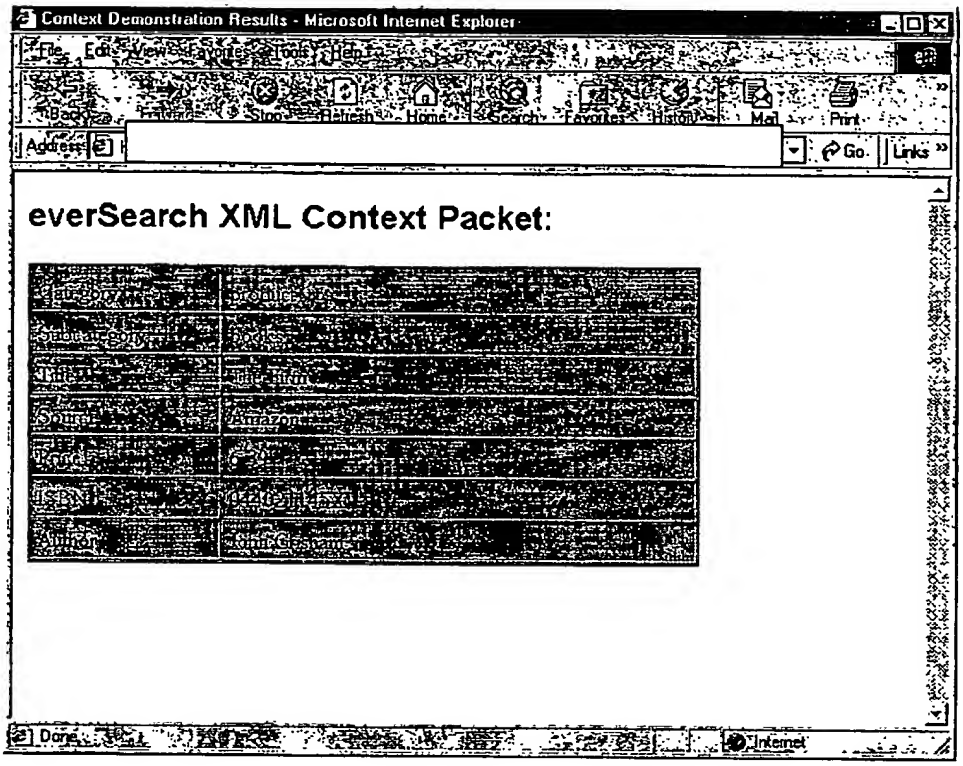

Context Demonstration Results - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address
Go Links
everSearch XML Context Packet:
Done
Internet

Figure 4.

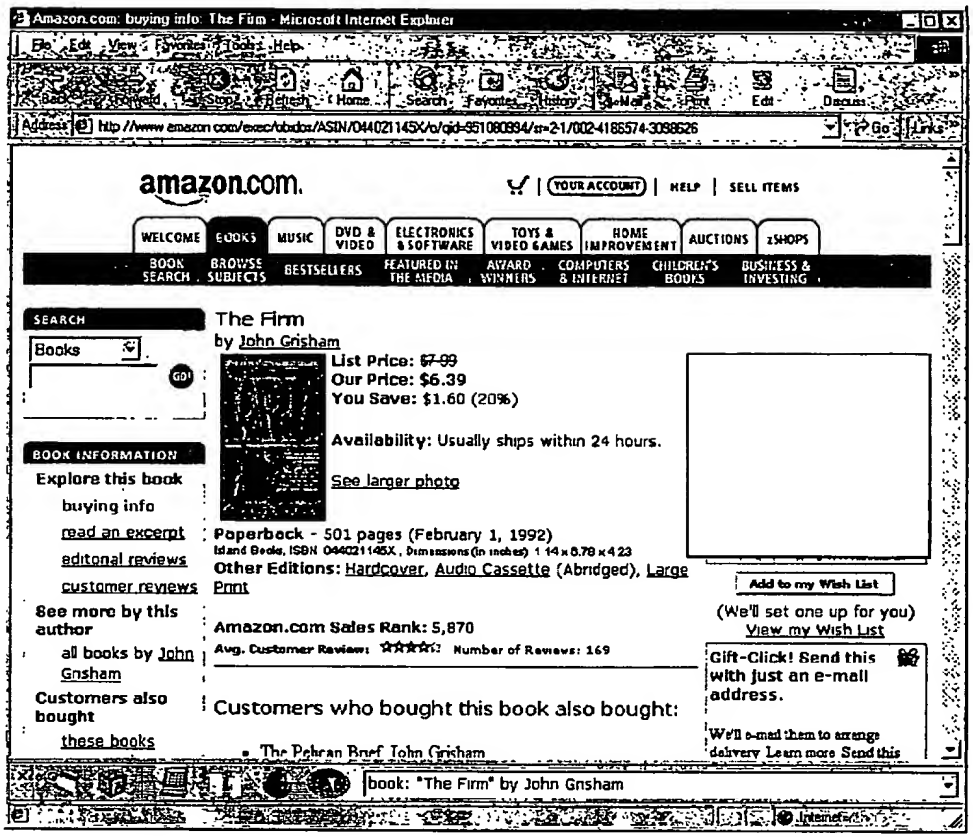

Amazon.com: buying info: The Firm - Microsoft Internet Explorer
amazon.com.
YOUR ACCOUNT | HELP | SELL ITEMS
WELCOME
BOOKS
MUSIC
DVD & VIDEO
ELECTRONICS & SOFTWARE
TOYS & VIDEO GAMES
HOME IMPROVEMENT
AUCTIONS
zSHOPS
BOOK SEARCH
BROWSE SUBJECTS
BESTSELLERS
FEATURED IN THE MEDIA
AWARD WINNERS
COMPUTERS & INTERNET
CHILDREN'S BOOKS
BUSINESS & INVESTING
SEARCH
Books
BOOK INFORMATION
Explore this book
buying info
read an excerpt
editorial reviews
customer reviews
See more by this author
all books by John Grisham
Customers also bought
these books
The Firm
by John Grisham
List Price: $7.99
Our Price: $6.39
You Save: $1.60 (20%)
Availability: Usually ships within 24 hours.
See larger photo
Paperback - 501 pages (February 1, 1992)
Other Editions: Hardcover, Audio Cassette (Abridged), Large Print
Amazon.com Sales Rank: 5,870
Number of Reviews: 169
Add to my Wish List
(We'll set one up for you)
View my Wish List
Gift-Click! Send this with just an e-mail address.
Customers who bought this book also bought:
book: "The Firm" by John Grisham

Figure 5.

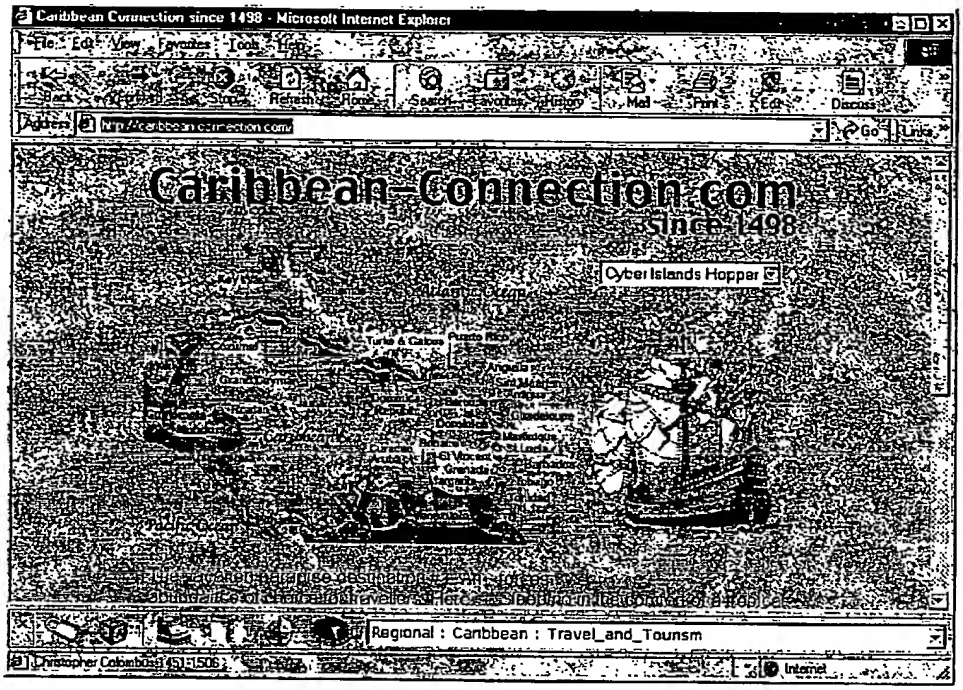

Caribbean Connection since 1498 - Microsoft Internet Explorer
Caribbean-Connection.com
Since 1498
Cyber Islands Hopper
Regional : Caribbean : Travel_and_Tounsm

Figure 6.

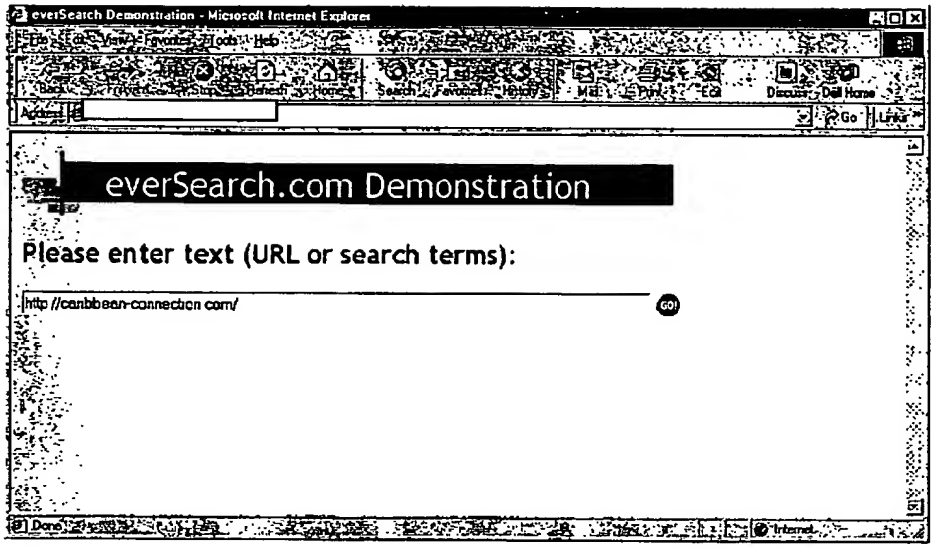

everSearch Demonstration - Microsoft Internet Explorer
everSearch.com Demonstration
Please enter text (URL or search terms):
http://caribbean-connection.com/
GO!

Figure 7.

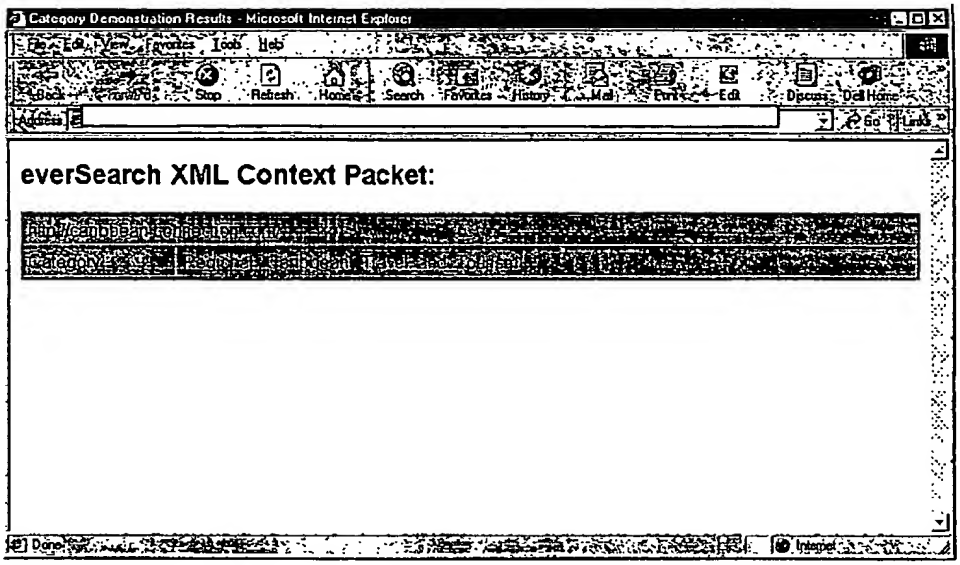

Category Demonstration Results - Microsoft Internet Explorer
everSearch XML Context Packet:

Figure 8.

METHOD AND SYSTEM FOR DISTILLING CONTENT

TECHNICAL FIELD

[0001] This invention relates generally to integrated systems and networks for processing information and more particularly to systems and methods for processing information available at various locations of disparate networks to form data groupings containing selected content.

BACKGROUND ART

[0002] Several new techniques and systems for processing and retrieving information have been developed with the proliferation of the Internet. Some of these developments are described in published documents.

[0003] In U.S. Pat. No. 5,937,407, an information retrieving apparatus is disclosed. The apparatus comprises a retrieve instruction executing means for executing a retrieve instruction based on a retrieval formula described based on an arbitrary schema, a schema conversion means for converting the retrieval formula into another retrieval formula according to another schema based on pregiven rules, and a schema management means for managing the rules for converting the retrieval formula into the other retrieval formula, wherein the retrieve instruction executing means retrieves desired information based on the other retrieval formula.

[0004] In U.S. Pat. No. 5,161,225, a persistent stream for processing time consuming and reusable queries in an object oriented database management system is disclosed. Time consuming and reusable queries are handled in an object oriented database management system by providing a persistent stream object class. The persistent stream object class is a subclass of the stream class which is typically provided to encapsulate the results of a query. The persistent stream class inherits all the attributes and methods of the stream class but also includes a "save" method for saving the results of a query. When a query names a persistent stream as it object, the query results are saved. The query may also be performed in background or batch mode. All time consuming and reusable queries are performed by sending a query message to the persistent stream class, to thereby automatically save the query results.

[0005] In U.S. Pat. No. 5,278,980, an iterative technique for phrase query formation and an information retrieval system employing the interactive technique are disclosed. An information retrieval system and method are provided in which an operator inputs one or more query words which are used to determine a search key for searching through a corpus of documents, and which returns any matches between the search key and the corpus of documents as a phrase containing the word data matching the query word(s), a non-stop (content) word next adjacent to the matching word data, and all intervening stop-words between the matching word data and the next adjacent non-stop word. The operator, after reviewing one or more of the returned phrases can then use one or more of the next adjacent non-stop-words as new query words to reformulate the search key and perform a subsequent search through the document corpus. This process can be conducted iteratively, until the appropriate documents of interest are located. The additional non-stop-words from each phrase are preferably aligned with each other (e.g., by columnation) to ease viewing of the "new" content words.

[0006] In U.S. Pat. No. 5,745,754, a sub-agent for fulfilling requests of a web browser using an intelligent agent and providing a report is disclosed. A World Wide Web browser makes requests to web servers on a network which receive and fulfill requests as an agent of the browser client, organizing distributed sub-agents as distributed integration solution (DIS) servers on an intranet network supporting the web server which also has an access agent servers accessible over the Internet. DIS servers execute selected capsule objects which perform programmable functions upon a received command from a web server control program agent for retrieving, from a database gateway coupled to a plurality of database resources upon a single request made from a Hypertext document, requested information from multiple data bases located at different types of databases geographically dispersed, performing calculations, formatting, and other services prior to reporting to the web browser or to other locations, in a selected format, as in a display, fax, printer, and to customer installations or to TV video subscribers, with account tracking.

[0007] In U.S. Pat. No. 5,877,759, and interface for user/agent interaction is disclosed. A user interface, for example for Internet and intranet agents, embodies the technical potential of automation and delegation into a cohesive structure. The invention also provides intelligent assistance to the client user interface and provides an interface that is centered on autonomous processing of whole tasks rather than sequences of commands, as well as the autonomous detection of contexts which require the launch of a process, especially where such context is time-based.

[0008] U.S. Pat. No. 5,761,496 describes a similar information retrieval system and method. The retrieval request input means **110** reads a retrieval request consisting of input keywords set up by the user as well as their importance degrees. The retrieval management section **120** causes the relation keyword generation section **121** and the retrieval expression generation section **122** to generate a retrieval expression by using background knowledge and retrieval parameters. The retrieval management section **120** causes the database management section to retrieve data from the database **160** based on a generated retrieval expression, causes the relation data acquisition section **124** to present a temporary retrieval result to the user, and causes the relevance database management section **123** to store user-instructed relation data into the relevance database **150**. The retrieval management section **120** changes the retrieval parameters based on this relation data, causes the retrieval expression generation section **122** to generate a mew retrieval expression, and causes the database management section **125** to retrieve data again. The retrieval result output section **130** outputs the final retrieval result. Thus, this system allows the user to reflect his retrieval strategy and background knowledge about data easily and precisely and to execute similarity retrieval efficiently on a trial and error basis, without a substantial increase in the retrieval time.

[0009] In U.S. Pat. No. 5,768,578, a user interface for an information retrieval system is described. An improved information retrieval system user interface for retrieving information from a plurality of sources and for storing information source descriptions in a knowledge base. The

user interface includes a hypertext browser and a knowledge base browser/editor. The hypertext browser allows a user to browse an unstructured information space through the use of interactive hypertext links. The knowledge base browser/editor displays a directed graph representing a generalization taxonomy of the knowledge base, with the nodes representing concepts and edges representing relationships between concepts. The system allows users to store information source descriptions in the knowledge base via graphical pointing means. By dragging an iconic representation of an information source from the hypertext browser to a node in the directed graph, the system will store an information source description object in the knowledge base. The knowledge base browser/editor is also used to browse the information source descriptions previously stored in the knowledge base. The result of such browsing is an interactive list of information source descriptions which may be used to retrieve documents into the hypertext browser. The system also allows for querying a structured information source and using query results to focus the hypertext browser on the most relevant unstructured data sources.

[0010] In U.S. Pat. No. 5,918,214, a system and method for finding product and service related information on the Internet are described. A novel system and method for finding product and service related information on the Internet. The system includes Internet Servers which store information pertaining to Universal Product or Service Number (e.g. UPC number) preassigned to each product and service registered in the system, with Uniform Resource Locators (URLs) that point to the location of one or more information resources on the Internet, e.g. World Wide Websites, related to such products or services. Each client computer system includes an Internet browser or Internet application tool which is provided with a "Internet Product/Service Information (IPSI) Finder" button and a "Universal Product/Service Number (UPSN) Search" button. The system enters its "IPSI Finder Mode" when the "IPSI Finder" button is depressed and enters the "UPSN Search Mode" when the "UPSN Search" button is depressed. When the system is in its IPSI Finder Mode, a predesignated information resource (e.g. advertisement, product information, etc.) pertaining to any commercial product or service registered with the system is automatically accessed from the Internet and displayed from the Internet browser by simply entering the registered product's UPN or the registered service's USN into the Internet browser. When the system is in its "UPSN Search Mode", a predesignated information resource pertaining to any commercial product or service registered with the system is automatically accessed from the Internet and displayed from the Internet browser by simply entering the registered product's trademark(s) or (servicemark) and/or associated company name into the Internet browser.

[0011] In U.S. Pat. No. 5,761,663, a method for distributed task fulfillment of web browser requests is described. A World Wide Web browser makes requests to web servers on a network which receive and fulfill requests as an agent of the browser client, organizing distributed sub-agents as distributed integration solution (DIS) servers on an intranet network supporting the web server which also has an access agent servers accessible over the Internet. DIS servers execute selected capsule objects which perform programmable functions upon a received command from a web server control program agent for retrieving, from a database gateway coupled to a plurality of database resources upon a single request made from a Hypertext document, requested information from multiple data bases located at different types of databases geograhically dispersed, performing calculations, formatting, and other services prior to reporting to the web browser or to other locations, in a selected format, as in a display, fax, printer, and to customer installations or to TV video subscribers, with account tracking.

[0012] U.S. Pat. No. 5,913,215 discloses an apparatus and method for identifying one of a plurality of documents stored in a computer-readable medium. The method includes the steps of prompting a computer-user to construct a search expression, then communicating the search expression to each of a plurality of search engines located at respective World Wide Web sites. Each of the plurality of search engines is prompted to concurrently identify a respective plurality of web pages containing text consistent with the search expression and to return a respective URL for each such web page identified. Redundant URLs returned by the search engines are filtered to obtain an initial set of web pages. Each of the initial set of web pages is downloaded and linguistically analyzed to automatically identify for the computer-user keyword phrases therein. The computer-user is prompted to construct a query expression in which one or more keyword phrases from the initial set of web pages is an operand. The query expression is then used to identify at least one web page of the initial set of web pages and the identified web page is presented to the user in the form of an abstract.

[0013] In U.S. Pat. No. 5,907,838, an information search and collection method and system are described. A method and apparatus in which category classes express information content categories that are defined based on object-oriented programming. The information items that are to be collected for each category are set as properties, and an information acquisition method or information process and treatment method is described for each property. After a request input from a user has been converted into a request input format that the system can understand, the request input is classified into category classes, searching is performed, and the information items the system outputs are displayed using the properties of the classes to which the request input belongs. Information searching and collection is accomplished on the basis of the contents described by the methods, and the information is output as comprehensive information in accordance with the request input of the user.

[0014] In U.S. Pat. No. 5,793,964, a web browser system is described. A World Wide Web browser makes requests to web servers on a network which receive and fulfill requests as an agent of the browser client, organizing distributed sub-agents as distributed integration solution (DIS) servers on an intranet network supporting the web server which also has an access agent servers accessible over the Internet. DIS servers execute selected capsule objects which perform programmable functions upon a received command from a web server control program agent for retrieving, from a database gateway coupled to a plurality of database resources upon a single request made from a Hypertext document, requested information from multiple data bases located at different types of databases geograhically dispersed, performing calculations, formatting, and other services prior to reporting to the web browser or to other

locations, in a selected format, as in a display, fax, printer, and to customer installations or to TV video subscribers, with account tracking.

[0015] U.S. Pat. No. 5,913,214 describes a system for querying disparate, heterogeneous data sources over a network, where at least some of the data sources are World Wide Web pages or other semi-structured data sources, includes a query converter, a command transmitter, and a data retriever. The query converter produces, from at least a portion of a query, a set of commands which can be used to interact with a semi-structured data source. The query converter may accept a request in the same form as normally used to access a relational data base, therefore increasing the number of data bases available to a user in a transparent manner. The command transmitter issues the produced commands to the semi-structured data source. The data retriever then retrieves the desired data from the data source. In this manner, structured queries may be used to access both traditional, relational data bases as well as non-traditional, semi-structured data bases such as web sites and flat files. The system may also include a request translator and a data translator for providing data context interchange. The request translator translates a request for data having a first data context into a query having a second data context which the query converter described above. The data translator translates data retrieved from the data context of the data source into the data context associated with the request. A related method for querying disparate data sources over a network is also described.

[0016] In U.S. Pat. No. 5,931,907, a software agent for comparing locally accessible keywords with meta-information and having pointers associated with distributed information is disclosed. A system for accessing information stored in a distributed information database provides a community of intelligent software agents. Each agent can be built as an extension of a known viewer for a distributed information system such as the Internet World Wide Web. The agent is effectively integrated with the viewer and can extract pages by means of the viewer for storage in an intelligent page store. The text from the information system is abstracted and is stored with additional information, optionally selected by the user. The agent-based access system uses keyword sets to locate information of interest to a user, together with user profiles such that pages being stored by one user can be notified to another whose profile indicates potential interest. The keyword sets can be extended by use of a thesaurus.

[0017] At present, it is very common for users of the Internet to manually search for relevant information using search engines such as those available at Internet locations such as Yahoo! (www.yahoo.com) or AltaVista (www.altavista.com). Other Internet services, such as those available at Ask Jeeves (www.ask.com) or MetaCrawler (www.metacrawler.com), are configured to use a single query to search more than one other service for relevant information based upon the user's manually entered query. While each of these services may be useful, each requires the manual entry of information. With manual entry techniques, users spend time experimenting with entry keywords and looking through long lists of available content which may or may not be relevant or useful.

[0018] Some other providers, such as FlySwat (www.flyswat.com), have attempted to bypass this manual information entry step by analyzing all or most of the text content of a page which a user is visiting. While such techniques may bypass the manual entry step, they may also return to the user content which is not particularly relevant or desirable because they generally have no means for distilling the content of a visited page into associated pieces of information which may be used to search for and return to the user content which is more likely to be useful and relevant.

[0019] There is a need for a system and method for efficiently distilling the content of visited pages into meaningful subgroups of information. Distilled content may be used for various purposes such as reduced content browsing and focussed background searching.

SUMMARY OF THE INVENTION

[0020] This is a method for distilling content from an Internet location. In one variation, the inventive method comprises comparing URL information associated with an Internet location with a rule trigger in a manner which compares characters comprising the URL with rule trigger characters which comprise the rule trigger to find a match. A rule, or rule algorithm, is then executed based upon the associated match to extract subexpressions from HTML and URL information of the Internet location and compile the subexpressions into a distilled data packet, or datagram.

[0021] In another variation, the inventive method comprises comparing the characters of URL information associated with an Internet location with the characters of each of a set of rule triggers to calculate scores for the comparisons based upon numbers of matches and weights assigned to each. The highest scoring rule having a score greater than some threshold score is applied as the default rule.

[0022] In another variation, the inventive method comprises comparing the characters of URL information associated with an Internet location with the characters of each of a set of rule triggers to calculate scores for the comparisons based upon numbers of matches and weights assigned to each. A rule algorithm associated with the rule trigger with the greatest score which is greater than or equal to a threshold score is executed to extract subexpressions from the HTML and URL information associated with the Internet location and compile the subexpressions into a datagram.

[0023] In another variation, the inventive method comprises downloading a first content-known page having first content comprising a first value for a keyword or tag. A first minimum regular expression is formed to extract the first value for the first keyword. A second content-known page is then downloaded. The second content-known page comprises a second value for the keyword. A second minimum regular expression is then formed to extract the second value for the keyword. The first and second minimum regular expressions are compared and a determination is made regarding which one better extracts values for the keyword.

DETAILED DESCRIPTION

[0024] A key aspect of each variation of this invention is the distillation of information associated with an Internet location to which the user has browsed using various algorithms operating in the background to produce a linked grouping of distilled pieces of information (hereinafter a "datagram") which may be used in various ways to help the

user. The invention comprises techniques for leveraging the inventive datagram creation process in other information processing and transmission processes such as reduced display browsing, datamining, and selected content provision.

[0025] The Internet is a collection of information storage devices and processors disparately located and connected electronically to each other by network conduits comprising physical elements, such as fiber optic cables, or wireless technology which enables devices to communicate without physical contact. Users of the Internet typically find information using browser software, such as Microsoft Internet Explorer or Netscape Navigator, which is configured to navigate a text-based version of the Internet called the World Wide Web (hereinafter "the web") by reading and downloading information such as text, which is generally made available by programmers in HTML (hypertext markup language) format.

[0026] Browser software typically is installed on a user's local information system, such as a personal computer, personal data assistant ("PDA"), cell phone, or similar device which generally has temporary memory, such as random access memory (or "RAM"), more permanent storage capacity, such as that provided by a hard disk drive, a locally installed information processing device such as a Pentium(TM) microprocessor, and an Internet connectivity device such as a modem. The Internet connectivity device generally is configured to establish electronic contact between a local information system and a remotely located device, such as a modem bank of an Internet service provider, which bridges the electronic connection of the local information system to other systems connected via the Internet. In the case of some devices such as digital cell phones, an Internet connectivity device may not be required, as the digital cell phone may contact the Internet directly or indirectly without the use of a modem, depending upon the cell phone network configuration.

[0027] When a user browses the web from a local information system, information from remote systems is transferred (or "downloaded") from the remote systems to his local system, often in HTML format. The user's locally installed browser software is configured to display a web "page" based upon the content of the downloaded information, which may comprise text, pictures, movie clips, music clips, and other elements known in the art of web design.

[0028] A key aspect of browsing the web is telling the browser software where to seek information which may subsequently be downloaded to the user's local information system. Browser software, such as Microsoft Internet Explorer and Netscape Navigator, is generally configured to provide the user with several options for navigating. Depending upon the content programmed into the particular web page, the user may be provided with "links" which are configured to download content associated with such links to the user's computer. Each link is associated with a Uniform Resource Locator, or URL, which is a brief instruction set pointing to the desired information. Links are generally displayed on a web page using a standard bold/underlined format in a particular color, such as blue, designed to communicate to the user that he will receive content associated with the link by "clicking" on the link using his pointing device (such as a mouse or other pointing device known to those skilled in the art of personal information system design).

[0029] Most browser software also allows users to directly input URL text for download of the associated information without the step of clicking on a link.

[0030] When a user uses a typical "search engine", such as that found at www.altavista.com, to find desired content, he generally enters text keywords, activates a search, and receives a list of links in return, the links being associated with URLs.

[0031] In short, browsing the web comprises using a URL to download information, generally comprising text, from a remote information system to a local information system.

[0032] Datagrams:

[0033] This invention comprises a method and apparatus for analyzing the content of URLs and HTML pages to form distilled data packets or "datagrams" comprising portions of the URL or HTML content selected according to a set of rules. A datagram is a description of the content of a web page. It may contain a complete description of all of the contents of the web page, but typically contains only the most essential pieces of information to describe the primary context of the web page. Datagrams generally are formatted in XML, a format which allows the data contained within to be highly structured and unambiguous. Datagrams generated by the inventive system may be stored, in database format, for example, remotely or locally and used for various purposes, such as searching for content on the web based upon datagram content, or enabling certain forms of reduced display browsing. In one variation, a datagram comprises a grouping of tag/value pairs. In another variation, a datagram may comprise portions of a URL.

[0034] Datagram Formation:

[0035] To form a datagram, URL or HTML information must somehow be captured and analyzed. In one variation, this is accomplished using a piece of software known to those skilled in the art of computer software development as a "plug-in". The plug-in is configured to add new functionality to the existing browser software. In this variation, a plug-in is configured to "handshake" with the browser software in a manner wherein it receives URL and HTML information from the browser software and may cause the browser to send out URLs to download certain information. The plug-in also is configured to process incoming URL and HTML information using software rules which may be resident within the plug-in or located remotely on another information system such as a server. In another variation, datagram formation may occur entirely on a remote information system such as a server. Entirely server-based variations may be preferred for certain applications of the inventive datagram formation techniques, such as datamining and reduced display browsing.

[0036] Having a plug-in or other infrastructure for receiving, comparing, and sending URL and HTML information is only a portion of the preferred datagram formation process. In order to extract or distill content from a web page into a datagram, the invention must have some technique for determining what in particular to extract from the available information.

[0037] Rules:

[0038] In the preferred variation, "rules" dictate what content will comprise the datagram for a particular page. Since many web pages are different in that they have different information at different locations on their pages, different rules are needed for different pages.

[0039] For example, if the user is looking for a video and browses to an Amazon.com web page using the URL "http://www.amazon.com/exec/obidos/6302935148/ref=ed_oe_v hs/103-5023833-6266201", his local browser will download a web page comprising a video title, a purchase price, an image, and the star of the video movie. In this example, the title is the first item in the upper left corner of the page. An image of the video cover is below the title. The price is next to the "$" symbol, and the star, Tom Cruise, is next to the term "Starring:".

[0040] Finding the same item at Blockbuster (using, for example, the URL "http://www.blockbuster.com/mv/detail-.jhtml?prodid=97402& catid=500") results in a similar but different page with the image in the upper left corner, the title to the right of the image, the stars next to "Actors:", and the price next to the "$" symbol.

[0041] If it is desirable to distill the content of the two pages associated with the two aforementioned URLs, say perhaps into movie title, lead actor, price, and vendor, for comparison purposes, for example, then two different rules will be needed: one rule configured specifically to extract this information from the Amazon.com page, and the other configured specifically to do the same from the Blockbuster page. To select which rule or rules should be executed for a given web page, the preferred variation utilizes "rule trigger logic".

[0042] In the preferred variation, the URL of a web site which the user is viewing is sent to the plug-in and is analyzed by this preprogrammed rule trigger logic. The rule trigger logic, preferably coded as part of the software running locally due to speed advantages, is configured to examine the content of the text which comprises the URL, and to execute specific rules logically related to specific triggers in the trigger logic. For example, if the user is at the URL "http://www.amazon.com/exec/obidos/6302935148/ref=ed_oe_v hs/103-5023833-6266201", the preferred variation of the plug-in software would receive this URL as text after a "document complete" signal from the browser software and would analyze the whole phrase as well as subportions thereof. The rule trigger logic, preferably character string comparison logic, a set of "if-then" statements or a "hash table lookup" for comparing character string portions, or similar coding technique known to computer programmers, would be executed to analyze the URL. The object of the rule trigger logic is the find executable rules which are applicable to the particular site and execute these rules. Using the aforementioned pages to demonstrate, the rule trigger logic will be configured to analyze "http://www.amazon.com/exec/obidos/6302935148/ref=ed_oe_v hs/103-5023833-6266201" and make note of phrases such as "amazon.com" and "vhs" so a rule specifically designed to extract the proper distilled information from an Amazon-.com videotape product page could be selected and executed. In other words, the phrases "amazon.com" and "vhs" within the same URL may "trigger" a specific rule.

[0043] The subprocess of triggering rules may be simple or complex, depending upon the complexity of the rule trigger patterns being analyzed. For example, a trigger pattern may operate somewhat like "if A, then execute rule #1". This requires only very simple analysis to determine if "A" exists within the content of the page. If it is there, "rule #1" is executed. On the other hand, a trigger pattern may operate somewhat like "If A, and B, and C, and D, and E, then execute rule #2". In this case, "rule #2"has more specific requirements and may not be executed as often as "rule #1" because each of "A" through "E", inclusive, must be present. If the rule trigger logic is analyzing 100 similarly detailed rule trigger patterns simultaneously to determine which rules to execute given the content of a page, a significant amount of processing may be required. The creation of rule trigger patterns may occur manually using experimentation, or may occur automatically, as is described below.

[0044] The rules, preferably "regular expressions" or XML objects, each of which are known to programmers and described at online sites such as www.w3.org or in publications such as *Learning Perl* (O'Reilly & Associates, Inc., 1993), may generally be described as comprising pattern matching objects configured to extract phrases known as subexpressions from both the URL and HTML content associated with the downloaded page. Rules may be implemented in any form of computer instruction (binary, interpreted, or data-driven, for example). A rule might extract subexpressions from not only the page content, such as the movie title and product price, but also from the URL itself, such as the phrase "amazon.com". It is the extracted subexpressions which become portions of the datagram.

[0045] In the preferred variation, a datagram comprises at least one set of "tag/value pairs". The goal of the rules is to provide values to match with the tags in a completed datagram. For example, a datagram shell for a rule configured to distill the content of an Amazon.com videotape product page may comprise four tags: title, star, price, and vendor. When the proper rule executes, preferably locally using the plug-in as a conduit for the URL and HTML information, it will return subexpression values to match the three tags and the result will, hypothetically, be the following tag/value pairs: title/"The Firm", star/"Tom Cruise", price/"19.95", vendor/"amazon.com". Another rule configured to extract similar subexpressions from Blockbuster pages could return a datagram with the following tag/value pairs: title/"The Firm", star/"Tom Cruise", price/"19.95", vendor/"blockbuster.com". One can see that a price comparison between the two vendors could be accomplished quite easily having these two datagrams; indeed, price comparison is one of the many objects of this invention.

[0046] Default Rules:

[0047] In accord with the discussion above, after the rule trigger logic is used to determine which rule should be executed, the proper subexpressions may be extracted from the content comprising the web page. In situations where no specific rule match is found after the rule trigger logic is applied, a default rule may be selected or developed to extract selected subexpressions despite the failure to find a specific rule match. Several varations of default rule based datagram formation, or "default distillation", have been developed.

[0048] In one variation of default distillation, each available rule may be executed upon the content associated with

the web page (URL information, HTML text content, etc.). The results of the each rule execution are scored, based upon the number of rule trigger matches and a weight assigned to each match which is related to the descriptiveness of the particular match (ISBN number, for example, an international number associated with a specific book, would be highly weighted). The rule having the highest score above some threshold number would be assigned to the particular page as the default rule and the results of the rule execution would become the distilled data for the page.

[0049] In another variation, each of the rules may be executed, and a hybrid datagram returned containing the value content associated with each matching key/value pair having a weight over a threshold amount.

[0050] In another variation of default distillation, the content associated with the web page (URL information, HTML text content, etc.) is searched for "known" values, which are associated with tags. A database of known tags/value pairs and groupings thereof is stored either on the local information system or on a remote system. Within each grouping, each of the known tag/value pairs is assigned a weight, depending upon it's usefulness in identifying something from the page. For example, if a user is looking for a book at Amazon.com, the ISBN tag, associated with the book's ISBN number, would be assigned a relatively high weight. A score for a grouping of tag/value pairs would be calculated as the number of tag/value pair matches with a particular page, influenced by the weight of each match. The highest scoring grouping, above some threshold score, would be selected and the matches within this grouping would comprise the datagram. If, for example, the user came upon a page and the rule trigger logic was not able to identify and execute a specific rule particularly tailored for the page, but the default rule process was able to identify an ISBN value and a $ value, the textual content adjacent the "$" and "ISBN" tags, or the values, could be extracted. Having these two tag/value pairs at the same page is somewhat indicative that the user is at a book page and the price is given on the page. The ISBN number and the price information may be stored as distilled content of the page. If a significant list of groupings with similarly high scores results, the tag/value pairs of the groupings are analyzed to develop categorical information which may be returned as the datagram content. For example, if a large list of high scoring groupings is returned from the analysis, each of which has an ISBN number as a tag/value pair, it may be decided that the user is examining a book page, and book-related categorical content may be returned as the datagram content.

[0051] In another variation known as "reverse lookup", the URL for the particular page is sampled and analyzed by comparing the text comprising it with elements of a directory database which may be locally or remotely resident. The directory database is comprised of keywords from the titles of various hierarchy branches within directories available on the web, such as those available at Yahoo! Using the database of directory keywords, the closest match between the text comprising the URL and the directory keyword text may be found, and subsequently the category information associated with the best match directory hierarchy branch may be used to populate the datagram for the particular page. A directory database is typically comprised of 1) a category tree 2) a list of urls and possible descriptions, titles, etc. as leaves of the tree. Example of a branch is Top/Shopping/Clothing, example of a leaf is (www.gap.com/ "Clothing Store"). We lookup www.gap.com (or a subportion of a url), and return Shopping:Clothing. If the URL is listed in more than one branch, the invention returns the best match directory hierarchy branch, as is stated above.

[0052] Automated Rulebuilding using Seed Data:

[0053] In another variation, a specific rule may be created automatically using a database of known "seed data". This procedure works similarly for correcting existing specific rules which fail to properly execute for some reason, such as a formatting change at a previously known page such as the product pages at Amazon.com.

[0054] In this variation, a local or remote database contains "seed" content from various web pages matched to keywords such as "author", "title", or "ISBN". This database is used as a source of "seed data" for building new rules. An example is helpful for describing this variation. Assume the User is at JoesBooks.com, a little known web site for books. When the User goes to a product page at JoesBooks.com, the rule trigger logic (described above) finds no direct matches based upon the URL information and is unable to execute a specific rule because none exist in the rule database, which may be local or remote on a server, for JoesBooks.com product pages. The database contains datagram information for seed books, such as "John Grisham, The Firm" and "Michael Crichton, Sphere" comprising their respective titles, authors, and ISBN numbers. The rule creation logic must next determine how to get to the product pages for seed books. This generally comprises finding a "submit" box on a page, navigating a product tree within the web site, or, as is preferable, inserting the product name or portions thereof into a query string, generally by adding such text to the URL as is known in the art of internet querying. At this point, the rule creation logic should have adequate means to get to the "John Grisham, The Firm" book product page, for example—and this is precisely what happens: the specific product page for "John Grisham, The Firm" is found at JoesBooks.com.

[0055] Next the content of the product page is downloaded, locally or to a remote server for processing. For the purposes of this example, this process is repeated for other known books, such as "Michael Crichton, Sphere". The process is repeated more times if the product pages at JoesBooks.com are less highly correlated than many other typical product pages are (see, for example, the product pages of Amazon.com; they are highly correlated in format). In one variation, the process is repeated the same number of times for any site—a number which affords a high degree of certainty that any variance within a sites product pages has been covered. With a relatively homogeneous site, in terms of product or item page formatting, 25 or so cycles is probably enough information to create a successful specific rule. Techniques for directly assessing the correlation of pages of a web site are known in the art of datamining and internet programming. A key aspect of this format correlation: the downloaded pages have a high correlation of quite a few things, and some key things which always differ upon comparison.

[0056] The content downloaded from each page is then analyzed. First, there must be a determination of what keywords or tags will be required of the rule. In this book example, assume that it is necessary that the rule be able to

extract "Author", "Title", and "ISBN". Starting with "author", the rule creation logic will search the downloaded content of the "John Grisham, The Firm" page and will create a separate minimum regular expression, preferably, to extract each occurrence of "John Grisham". If "John Grisham" occurs three times on the JoesBooks.com page for that product, the three minimum regular expressions to may, for example look like:

[0057] #1: I books by <a href="[^"]">([,a-zA-Z0-9 '&:\.\])\[\(\)#\-]*[a-zA-Z0-9\(\)\]\[\]][]*<\a>≧"

[0058] #2:<img width=60 height=92 src="[^"]*" alt="([,a-zA-Z0-9 '&:\.\]\[\(\)#-]*[a-zA-Z1-9\(\)\]\[]]*Store" border="0">

[0059] #3:>"by <a href="^"]*">([,a-zA-Z0-9 '&\.\]\[\(\)#\-]*[a-zA-Z0-9\(\)\]\[\][]*<\a>"

[0060] Continuing with the subprocess for developing a rule or subpart thereof for properly extracting the "author" from a JoesBooks.com product page, the rule creation logic will analyze the content of the "Michael Crichton, Sphere" page and create minimum regular expressions for each occurrence of "Micheal Crichton" on the page, resulting, for example, in three occurrences and three minimum regular expressions:

[0061] #1: I search books for<a href="[^"]*">([,a-zA-Z0-9 '&:\.\]\[\(\)#\-]*[a-zA-Z0-9\(\)\[\(\)#\-]<\a><"

[0062] #2:>"by <a href="[^"]*">([,a-zA-Z0-9 '&:\.\]\[(\)#\-]*[a-zA-Z0-9\(\)\(\)\]\[\])[

[0063] #3>([,a-zA-Z0-9 '&:\.\]\(\)#\-]*[a-zA-Z0-9\(\)\[]) []*Store<

[0064] Note that in these expressions, the original text "John Grisham" and "Michael Crichton" has been replaced as appropriate with a regular expression to match any author for the sample set (e.g., ([,a-zA-Z0-9 '&:\.\]\[\(()#\-]*[a-zA-Z0-9\(\\][])). From this analysis of two pages, one can see that the third expression for the Grisham book is identical to the second expression for the Crichton book. This expression may be chosen as the candidate for extracting "author" from a JoesBooks.com product page. If this identity was not found, the next best choice for a minimal expression would be the merger of the first expression from each (e.g., I(books by|search books for)([,a-zA-Z0-9 '&:\.\]\[\(\)#\-]*[a-zA-Z0-9\(\)]\[]) []*<").

[0065] This expression is then applied across the sample set to see if it still works/returns correct results. If not, the next best set is chosen. Assuming the identical expression #3 from "John Grisham" and #2 from the "Michael Crichton" is reapplied across the entire sample set (only two are shown here) and succeeds, it is chosen.

[0066] This process is repeated for each basic datum one wants to extract from a page (e.g. redo for Titles, in this case "The Firm" and "Sphere" respectively, then for ISBN number, etc.). Note that additional heuristics may be applied to help the minimal expression generation by added rules about relations between each item/datum on a page the user is extracting (e.g. choose expressions where the datums found are close to each other, if more than one is found they must repeat, e.g. author,title, author,title, etc.).

[0067] Once all the expressions have been found, they are packaged together into a rule, and the rule associated with the common portion of the URL (e.g. www.JoesBooks.com/products/ . . .) with which the dataset is associated.

[0068] The rule may be generated in various format, such as java, jscript, or compact data form. An example of compact data form is as follows:

```
<rule>
    <extractionset language="regex">
      <extractionItem>
        <regex>>"by <a href='[^"]*">([,a-zA-Z0-9 '&:\.\]\[\(\)#\-]*[a-zA-Z0-9\(\)\]\[
]*</a>"</regex>
        <tag id="0">author</tag>
      </extractionItem>
      <extractionItem>
        <regex>>" <font size=0x3><b>([,a-zA-Z0-9 '&:\.\]\[\(\)#\-]*[a-zA-Z0-9\(\)\]\[
]*</b></font>"</regex>
        <tag id="0">title</tag>
      </extractionItem>
    ...
</rule>
```

[0069] Regardless of the form of the rule, its output as a datagram is typically the same: an XML packet where each datum's tag (in this case, author and title) is the tag of an XML element:

```
<datagram>
  <author>John Grisham</author>
  <title>The Firm</title>
</datagram>
```

[0070] Such an XML object is easily parsed and stored into the database as a grouping of tag/value pairs.

[0071] Generating and executing rules for sets of relatively homogeneous single item pages, such as the book product pages viewable at Amazon.com, is made relatively routine using these automatic rule generation techniques. Tables or lists of items on a single page, otherwise known as a "multiple product page" presents a more complex problem. To illustrate, imagine a web vendor called AllMedia.com which sells books, movies, DVDs, etc. If a user browses to a single product page for "John Grisham, The Firm" at AllMedia.com, the distilled content techniques

should be able to extract a datagram from the content available at the page. But the scenario wherein the same query to AllMedia.com returns a table having three different listings for "John Grisham, The Firm", one for the book format, one for the videotape, and one for the DVD, is different. A table has regularity just like a group of correlated product pages; the key difference is that single product pages are associated with one item, while multiple product pages are associated with more than one item, and it is unclear upon first glance how many items. If there is more than one item, additional information can be added to the datagram or database regarding "John Grisham, The Firm"—namely that it is available in other formats.

[0072] To test the relationship of the other items to the one which caused a rule to properly execute, a "trusted source" is used for benchmarking. The text information associated with each of the other items is sent in a query format to a trusted source, such as Amazon.com. If the other items (namely the DVD and videotape) are found at the trusted source to be associated with "John Grisham, The Firm", then the additional information, namely a "new tag/value pair" associated with the others in the grouping for "John Grisham, The Firm", may be added to the grouping on the database for future reference.

[0073] Use of Datagrams:

[0074] In the preferred variation, transfer of information between the user's browser software and the plug-in, as well as the production of datagrams using rule trigger logic and executed rules, is conducted in the background so the user may continue to browse the web. After a datagram is constructed, it is preferably sent to a datagram processing system, such as a server, using the Internet conduit with which the user is browsing the web. Sending the datagram information from the plug-in an outside system is accomplished using standard protocols known to Internet programmers, such as HTTP (hypertext transfer protocol). The datagram processing system may also reside on the user's local information system.

[0075] Having the distilled information from more than one location allows for high-speed processing and analysis: partially due to the distilled nature of the datagram information, and partially due to the advantage of having the tag/value pairs in one location with known formats. The ability to distill the content of web pages into datagrams may be leveraged as an enabling portion of one variation of the inventive system and method comprising reduced display browsing. The inventive techniques for distilling web page content may also be leveraged for datamining purposes.

[0076] Reduced Display Browsing:

[0077] Reduced display browsing enables users to browse the web using devices such as PDAs, cell phones, pagers, or even watches which have small display screens in comparison to more traditional computer monitors for which much of the browsing software was designed. Some local information systems and their related networks, such as digital cell phones and service available from Sprint PCS or the "Palm-7"PDA from Palm Computing and it's associated digital broadcast service, enable limited web browsing using a small, relatively low resolution liquid crystal display present on the telephone hardware. Since a typical web page contains more text than can be readably displayed on such a display, services such as Sprint PCS broadcast reduced versions of certain web pages for users to read and interact with.

[0078] For example, some cellular phone services allow users to check stock quotes or use certain search engine pages. They generally do not, however, allow users to freely browse the web because much of the distillation of content available on the pages supported by the service is done via direct data export from the particular pages which are supported. For example, a cellular service may have an agreement in place with a stock quote web page wherein the stock quote service transmits the distilled data desired by the cellular service to the cellular service for subsequent transmission to users on their cell phones or PDAs.

[0079] Datagram formation enables direct export of distilled content from a given web page after a rule is fired. The distillation may occur at the direction of the broadcasting service provider, or it may occur automatically as the user browses from his limited display information system.

[0080] Datamining:

[0081] Another usage of datagrams is for data mining applications (also known as "data warehousing"). In datamining applications, the user or operator generally is interested in capturing or "mining" certain key portions of content from a larger set available on a web page or other information repository. The formation of datagrams in accord with the present invention may be leveraged as a routine for "mining" key content from websites since they contain distilled versions of the web pages generally comprising the portions of these pages likely to be most relevant to a user interested in datamining. Datagrams contained structured data, preferably formatted in XML, which allows other applications such as datamining applications to easily capture and organize key information.

EXAMPLES

Example: Datagram Extraction

[0082] 1) Web pages are found and accessed by what is referred to as a "URL" or "Uniform Resource Locator". The URL http://www.amazon.com/exec/obidos/ASIN/044021145X refers to the following page shown in **FIG. 1**.

[0083] A sample of the actual text, or HTML, of this page is shown in **FIG. 2**.

[0084] This is only a small portion of the HTML text—the entire page as seen above contains far more text.

[0085] 2) In one embodiment of the invention, the URL as seen above can be submitted to a remote processing server. A visual description of doing this via the web may look like that in **FIG. 3**.

[0086] The server processes the URL and uses trigger logic to find what rule to execute on the returned content associated with this URL. The content (generally in HTML format) represented by this URL is downloaded, and the rule executed.

[0087] The server then responds with a datagram, preferably an XML packet, here visually laid out in HTML in **FIG. 4** for clarity.

[0088] The actual XML for the returned packet would look similar to:

```
<node>
  <Category>product</Category>
  <Subcategory>books</Subcategory>
  <Title>The Firm</Title>
  <Source>Amazon</Source>
  <Price>6.39</Price>
  <ISBN>044021145X</ISBN>
  <Author>John Grisham</Author>
</node>
```

[0089] Note two significant things which have occurred: 1) A huge amount of data, in this case a large amount of HTML data describing this particular page, has been reduced by a rule to the key pieces of distilled information; 2)

[0090] The distilled information has been packaged into a highly structured form, readable by both humans and machines. This technology is very useful for databasing, datamining applications, and reduced display devices such as cellular phones and PDAs, among other things.

Example: Browser Plug-in (or "Browser Companion") and Feedback to the User

[0091] 1) In this example, referring to **FIG. 5**, the user has installed a browser companion, powered by the inventive datagram creation technology, to work with the browser software. In this variation, the companion gives feedback to the User with a "toolbar" which can be seen at the bottom of the browser display.

[0092] 2) Here, the User is looking at the book: "The Firm" by John Grisham at Amazon.com.

[0093] 3) The browser companion displays for the User a feedback display regarding the particular page the User is looking at ("The Firm" by John Grisham). With the browser companion variation, the rules and rule triggers can be cached on the users machine (no immediate need to access the server if the rules are present). **(FIG. 5)**.

Example: "Reverse Lookup" Default Rule Situation

[0094] 1) In this example, the user has installed a browser companion, having datagram formation technology, to work with the browser software. This companion can be seen at the bottom of the browser, as a horizontal "toolbar."

[0095] 2) The user has gone to a new travel site, "Caribbean-connection.com"

[0096] 3) Assuming no specific rule exists, the system may do a reverse lookup through a directory database (e.g. the "open directory") to uncover the fundamental category for this site. This is novel in that such directory systems typically are used on site where the user enters a category, or traverses a category tree, to get to a site. Here, the user is already at a site, and the lookup is done to "reverse" the user to information regarding the appropriate category.

[0097] 4) The resulting category in the plugin browser are shown in **FIG. 6**.

[0098] 5) this category information may then be used to trigger appropriate related material.

[0099] The process can been seen visually by direct access to the knowledge base as shown in **FIGS. 7 and 8.**

[0100] 1) enter URL **(FIG. 7)**.

[0101] 2) The server responds with results **(FIG. 8)**

1. (form datagram using rules) A method for extracting content from Internet location information comprising:

a. comparing the URL information associated with an Internet location as well as subportions of said URL with a rule trigger in a manner which compares characters comprising said URL or subportions of said URL with rule trigger characters comprising said rule trigger to find at least one match;

b. executing a rule algorithm to extract subexpressions from the HTML and URL information associated with the Internet location and compile said subexpressions into a datagram.

2. (use HTML to for rule triggering) The method of claim 1 wherein the step of comparing further comprises comparing the HTML information associated with an Internet location with a rule trigger in a manner which compares characters comprising said HTML information with characters comprising said rule trigger to find at least one match.

3. The method of claim 1 wherein the rule is an XML object.

4. The method of claim 3 wherein the rule is a regular expression configured for extracting subexpressions from URL and HTML information.

5. The method of claim 1 wherein the step of comparing comprises using local plug-in software, which handshakes with local browser software operated on a local information system by said user, to import URL information associated with said Internet location from the local browser software and compare the URL and subportions thereof with the rule trigger.

6. The method of claim 1 wherein the step of comparing comprises using remote software running on a remote information system, which handshakes with local browser software operated on a local information system by said user, to import URL information from the local browser software and compare the URL and subportions thereof with the rule trigger.

7. The method of claim 5 wherein said rule is stored and executed on said local information system.

8. The method of claim 6 wherein said rule is stored and executed on said remote information system.

9. The method of claim 1 wherein the step of comparing between the URL or a subportion thereof and said rule trigger comprises using string compare logic to look for a match between the characters of the URL or subportion thereof and said rule trigger characters.

10. (when executing rule remotely, send URL from local, but download HTML directly at remote) The method of claim 8 wherein said URL information is sent to said remote information system from said local information system, while said HTML information is downloaded directly to said remote information system from said Internet location using said URL information.

11. (reduced display browsing) The method of claim 1 further comprising the steps of:

a. transmitting said datagram to a wireless information system; and

b. extracting said datagram to produce a reduced display view of the Internet location.

12. (default rule-1) A method for creating a rule algorithm for extracting selected content information from Internet location URL and HTML information comprising:

a. comparing the URL information associated with an Internet location as well as subportions of said URL with each of a set of rule triggers in a manner which compares characters comprising said URL or subportions of said URL with rule trigger characters of each rule trigger and calculates a score for each comparison based upon the number and weight of matches for a given comparison;

b. determining which rule trigger is the highest scoring rule trigger and determining that said highest score is greater than or equal to an application threshold score; b. executing a rule algorithm associated with the highest scoring rule trigger to extract subexpressions from the HTML and URL information associated with the Internet location and compile said subexpressions into a datagram.

13. (use HTML to for rule triggering) The method of claim 11 wherein the step of comparing further comprises comparing the HTML information associated with an Internet location with a rule trigger in a manner which compares characters comprising said HTML information with characters comprising said rule trigger to find at least one match.

14. The method of claim 11 wherein the rule is an XML object.

15. The method of claim 13 wherein the rule is a regular expression configured for extracting subexpressions from URL and HTML information.

16. The method of claim 11 wherein the step of comparing comprises using local plug-in software, which handshakes with local browser software operated on a local information system by said user, to import URL information associated with said Internet location from the local browser software and compare the URL and subportions thereof with the rule trigger.

17. The method of claim 11 wherein the step of comparing comprises using remote software running on a remote information system, which handshakes with local browser software operated on a local information system by said user, to import URL information from the local browser software and compare the URL and subportions thereof with the rule trigger.

18. The method of claim 15 wherein said rule is stored and executed on said local information system.

19. The method of claim 16 wherein said rule is stored and executed on said remote information system.

20. The method of claim 11 wherein the step of comparing between the URL or a subportion thereof and said rule trigger comprises using string compare logic to look for a match between the characters of the URL or subportion thereof and said rule trigger characters.

21. (when executing rule remotely, send URL from local, but download HTML directly at remote) The method of claim 18 wherein said URL information is sent to said remote information system from said local information system, while said HTML information is downloaded directly to said remote information system from said Internet location using said URL information.

22. (default rule-2) A method for creating a rule algorithm for extracting selected content information from Internet location URL and HTML information comprising:

a. comparing the URL information associated with an Internet location as well as subportions of said URL with each of a set of rule triggers in a manner which compares characters comprising said URL or subportions of said URL with rule trigger characters of each rule trigger and calculates a score for each comparison based upon the number and weight of matches for a given comparison;

b. determining which matches have a score which is greater than or equal to an application threshold score;

c. compiling the matches into a datagram.

23. [creating rules using seed data] A method for creating a selected content extraction rule for a series of correlated content pages comprising:

a. downloading a first content-known page having first content comprising a first value for a keyword;

b. forming a first minimum regular expression for extracting said first value for said keyword;

c. downloading a second content-known page having second content comprising a second value for said keyword;

d. forming a second minimum regular expression for extracting said second value for said keyword;

e. comparing said first minimum regular expression with said second minimum regular expression to make a determination regarding which of said first minimum regular expression or said second minimum regular expression better extracts values for said keyword.

* * * * *